Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks

WONG SHING MING
NATIONAL TSING HUA UNIV.

09 Mar. 2020

1. **Info. Of paper**
2. Related Work
3. System Pipeline - Training stage
4. System Pipeline - Inference stage (Authoring)
5. Results and discussion
6. Limitations, Failure Cases and Conclusion

Info. of paper

- Focus on generating a large-scale terrain with detail in the game using cGAN
- Research by INSA Lyon, Purdue University and Ubisoft
  - ➢Author: Guérin, Eric and Digne, Julie and Galin, Eric and Peytavie, Adrien and Wolf, Christian and Benes, Bedrich and Martinez, Benoit
- ACM Transactions on Graphics (proceedings of Siggraph Asia 2017)
  - ➢Times Cited: 19 (from Web of Science Core Collection at 23. Jan 2021)

1. Info. Of paper

2. **Related Work**

3. System Pipeline - Training stage

4. System Pipeline - Inference stage (Authoring)

5. Results and discussion

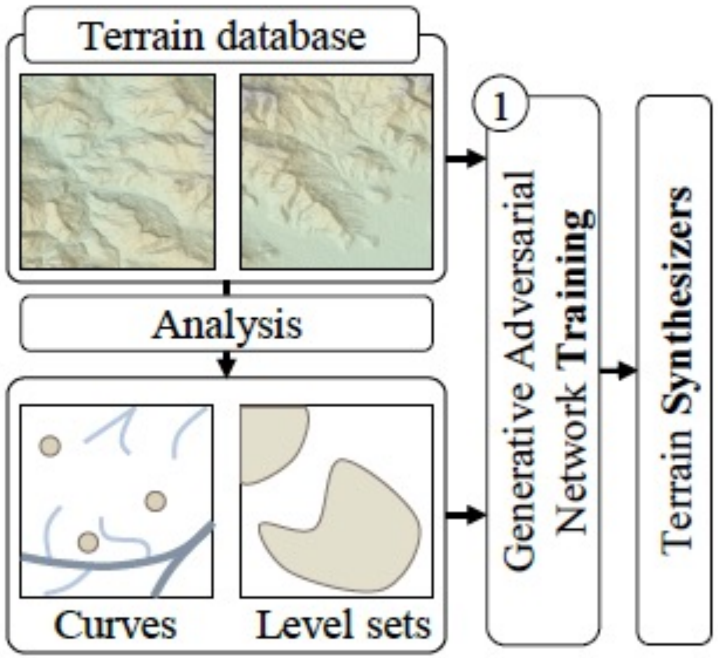
6. Limitations, Failure Cases and Conclusion

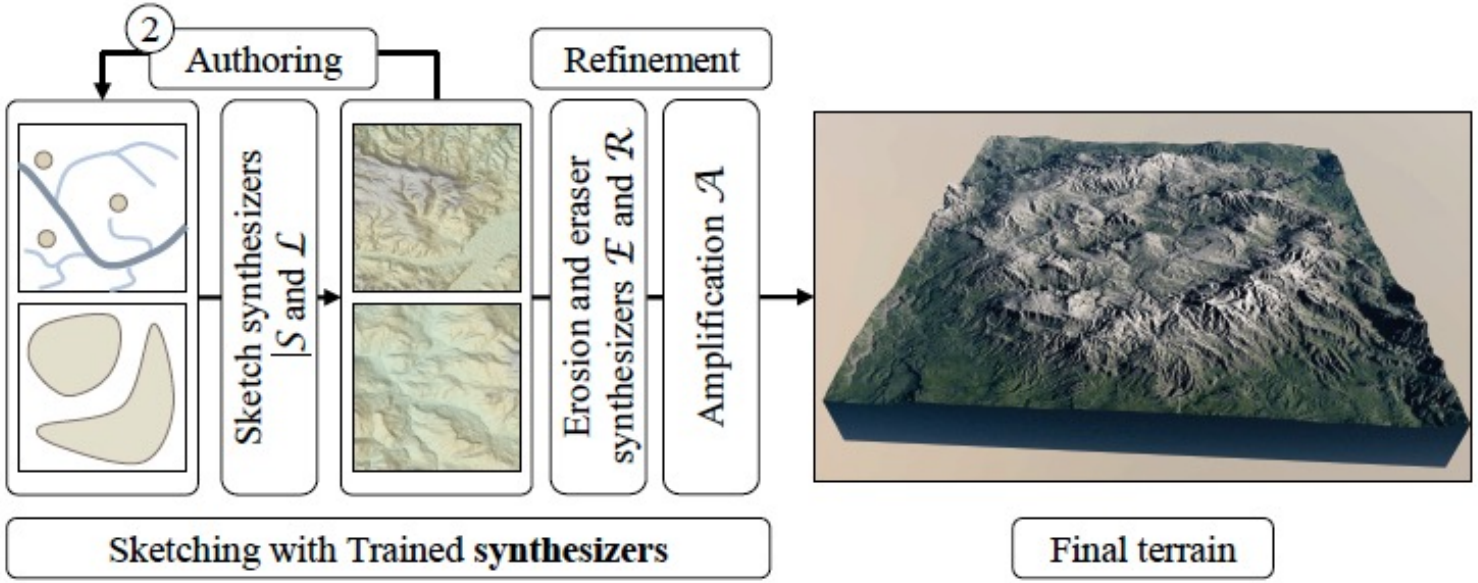
Related Work

Method	Pros	Cons
sketch-based	- easily control by user (high level of control)	- can't generate geologically correct outputs - tedious
simulation-based methods (erosion/hydrology-based algorithms)	- generating geologically correct models	- hardily control by user (lack user control) - computationally expensive
procedural methods	- generating the terrain fast - computationally efficient	- difficultly control
example-based methods	- generating large terrains using small examples	- provide low user-control - can't easily generate new features

The existing algorithms is that they cannot be easily applied to large-scale terrains.

The system pipeline





Training stage

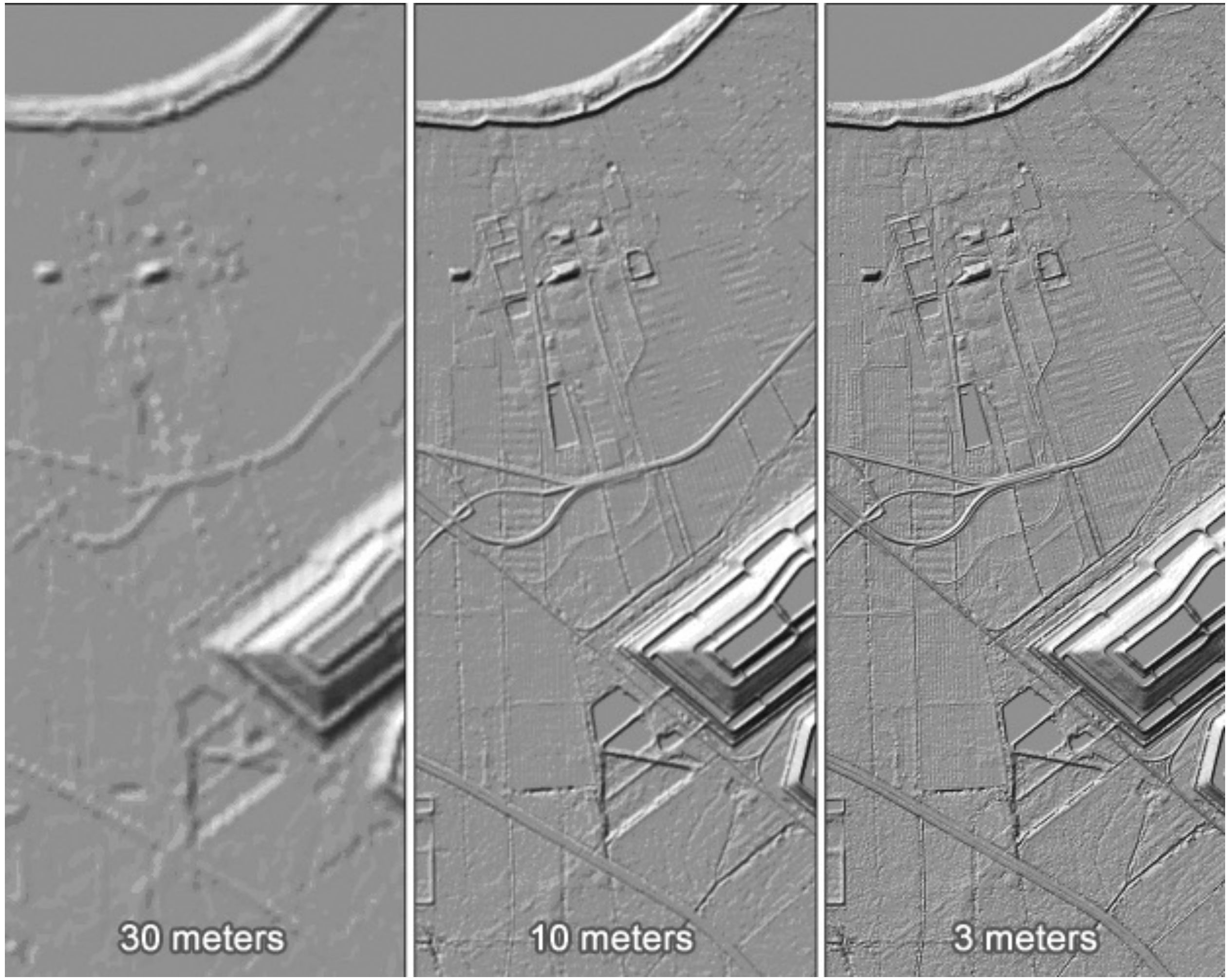
Inference stage (Authoring)

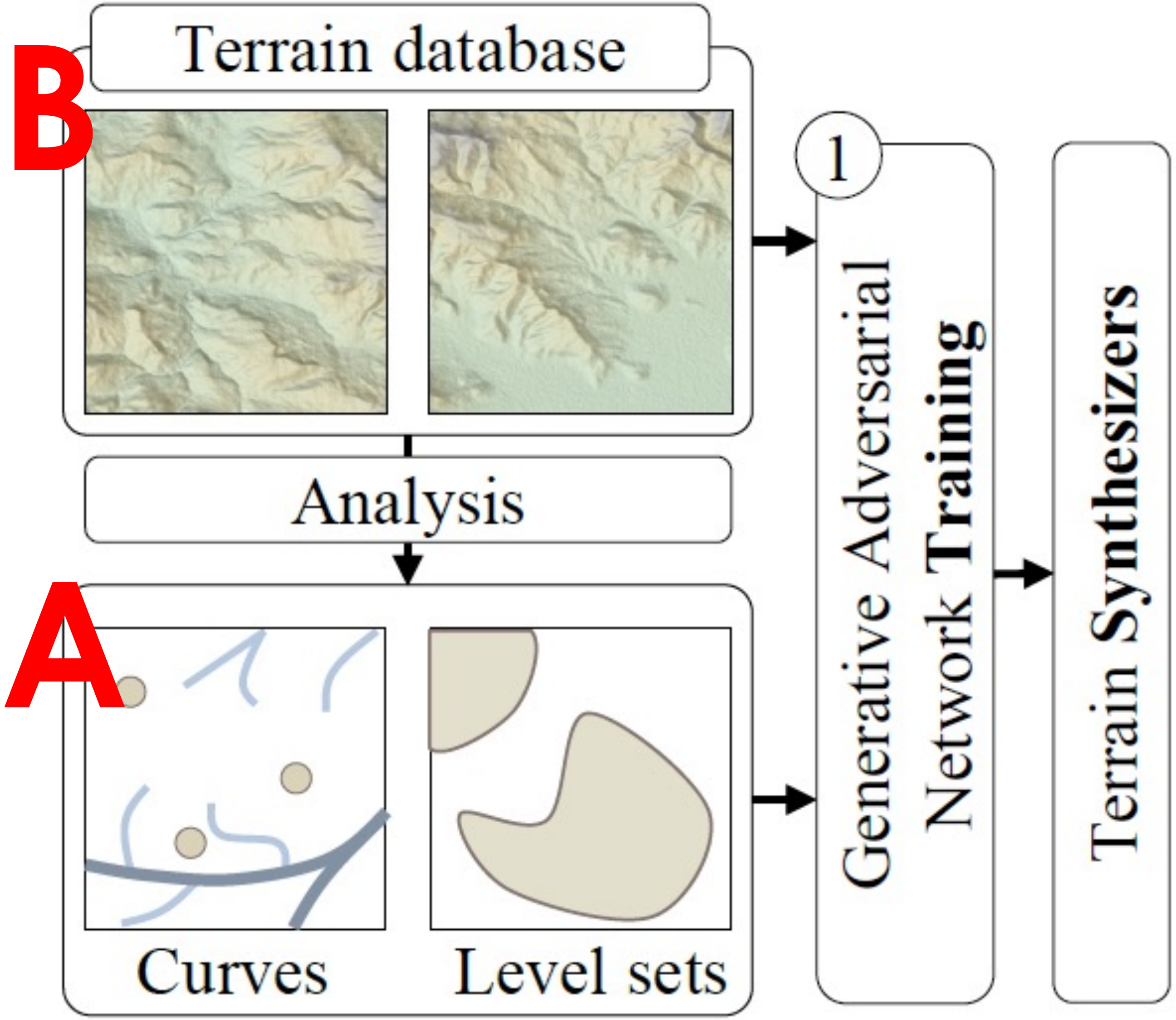
Digital Elevation Model (DEM)

X, Y = pixel

Pixel size -> small

Resolution -> high





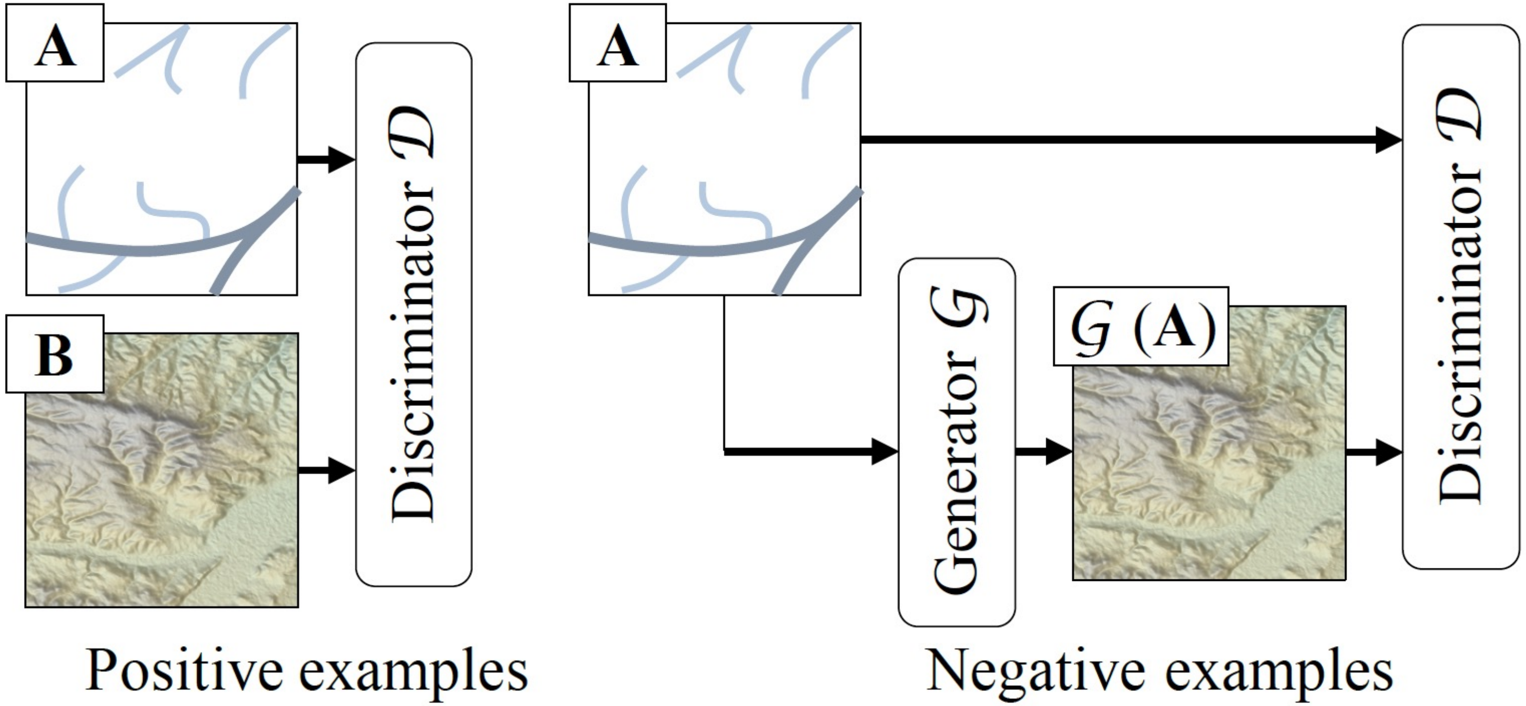
B real-world DEM

A its extracted sketch

$$\mathcal{D}(A, B) \rightarrow 1$$

$$\mathcal{D}(A, \mathcal{G}(A)) \rightarrow 0$$

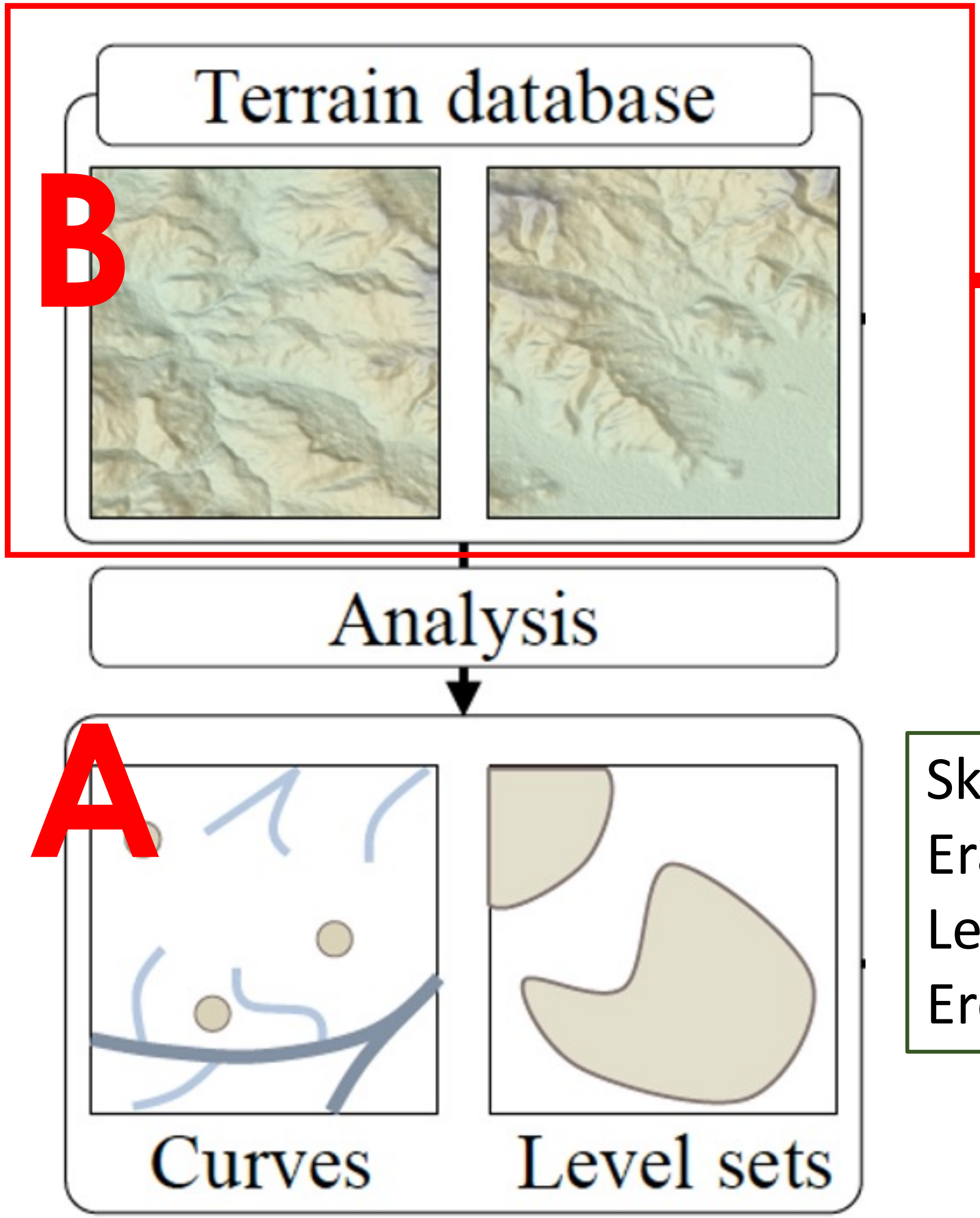
$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{(\mathbf{A},\mathbf{B})}[\log \mathcal{D}(\mathbf{A}, \mathbf{B})] + \mathbb{E}_{\mathbf{A}}[\log(1 - \mathcal{D}(\mathbf{A}, \mathcal{G}(\mathbf{A})))].$$



$$\mathcal{D}(A, B) \rightarrow 1$$

$$\mathcal{D}(A, \mathcal{G}(A)) \rightarrow 0$$

Fig. 4. Overview of the training of a cGAN: The discriminator $\mathcal{D}$ learns to classify between real and synthesized pairs, whereas the generator learns to fool the discriminator.



They present an automatic sketches generation process to build learning datasets

Sketch-to-terrain Synthesizer
Eraser Synthesizer
Levelset-to-terrain Synthesizer
Erosion Synthesizer

Input B and **computed A**

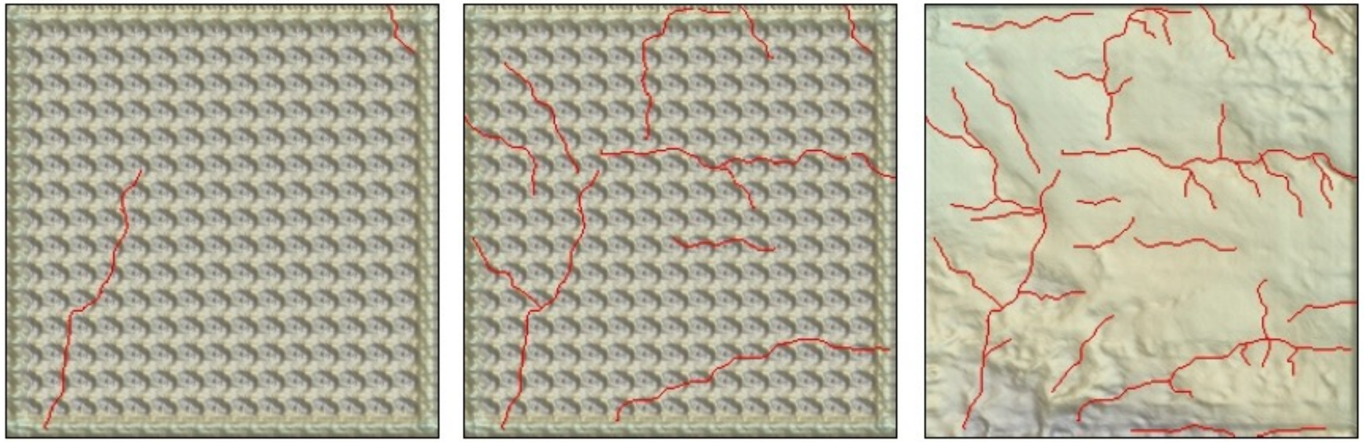
Training stage – Sketch-to-terrain Synthesizer -> River network

- Input: Altitude cues, rives, mountain rides
- River network
  1. using a modified river channel network algorithms [Tarboton et al. 1991]
  2. seed water over all the grid points of the terrain
  3. simulate flow using steepest descent D8 algorithms [O'Callaghan & Mark 1984]
  4. detecting the pixels with high water accumulation
     - ➢Using a stochastic direction at every step

Training stage – Sketch-to-terrain Synthesizer -> River network -> data pre-processing

- Does not correspond to a real user sketch of a river network -> user may need to draw a lot of strokes
- Solution:
  1. blurred the terrain and down-sampled before the flow simulation
  2. up-sampling after getting the resulting water accumulation (initial resolution of the rives)
- The training inputs that are coarse river directions
- Effect
  1. Not strictly respect constraints
  2. More flexibility in generator

Training stage – Sketch-to-terrain Synthesizer -> River network -> data pre-processing



No preprocessing

Over-constrained terrains

2 strokes 5 strokes 39 strokes

With preprocessing

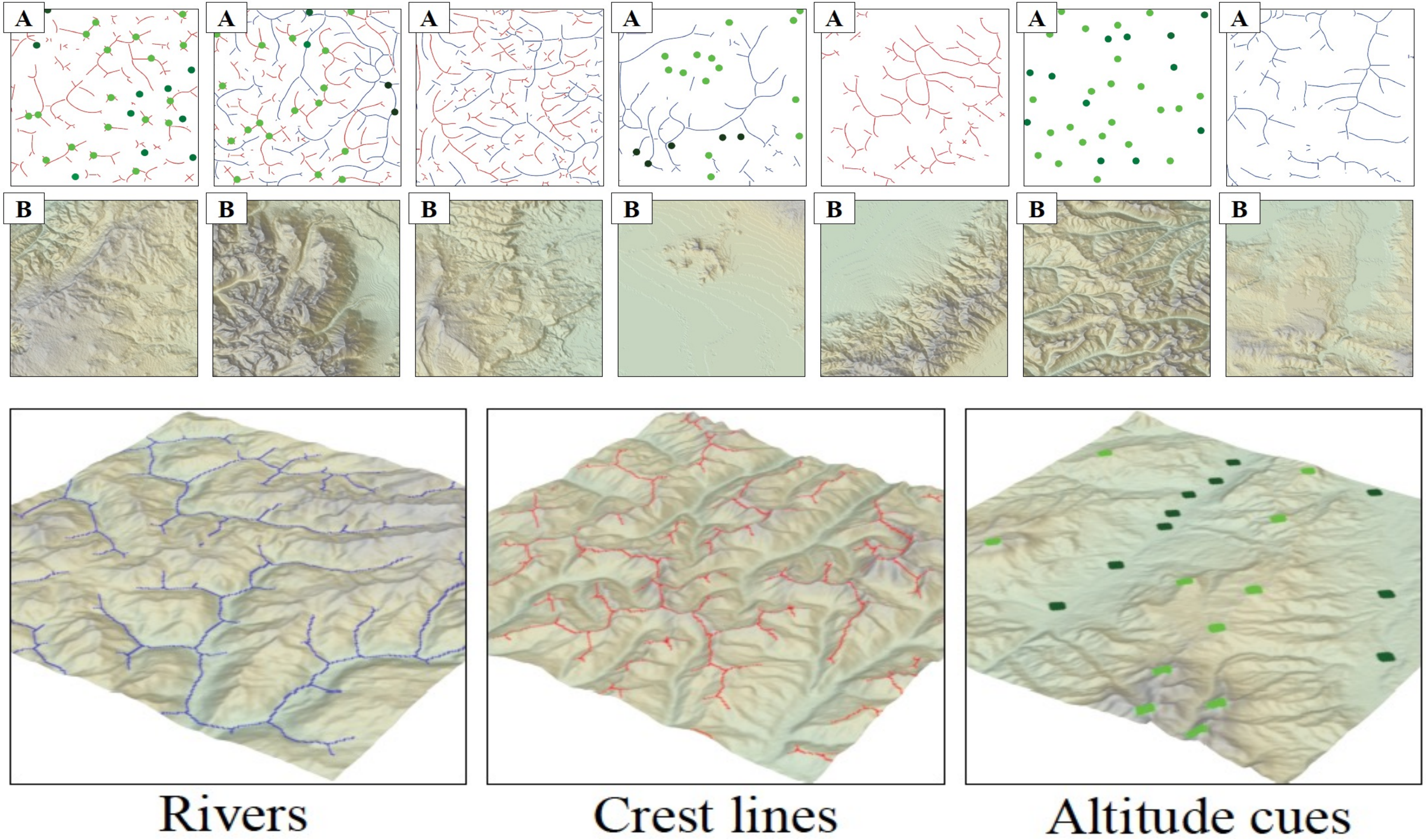
Blurred terrains

Training stage – Sketch-to-terrain Synthesizer -> Ridges

- Detected
  - ➢inverting the terrain
  - ➢applying the river detection algorithm (opposite to river detection)

Training stage – Sketch-to-terrain Synthesizer -> Altitude cues

- By a sparse set of peak and basin points over the terrain
  - Basin points are defined as point where the previous water flow accumulated above a chosen threshold
  - Peak points are defined by inverting the elevation of the terrain


A
B
Rivers
Crest lines
Altitude cues

Training stage – Levelset-to-terrain Synthesizer

- Provided as binary images
  - Include area in the terrain where the altitude is above a given percentile of the altitude distribution (60%)
  - Constructed by blurring the DEMs and thresholding the altitude at the provided percentile

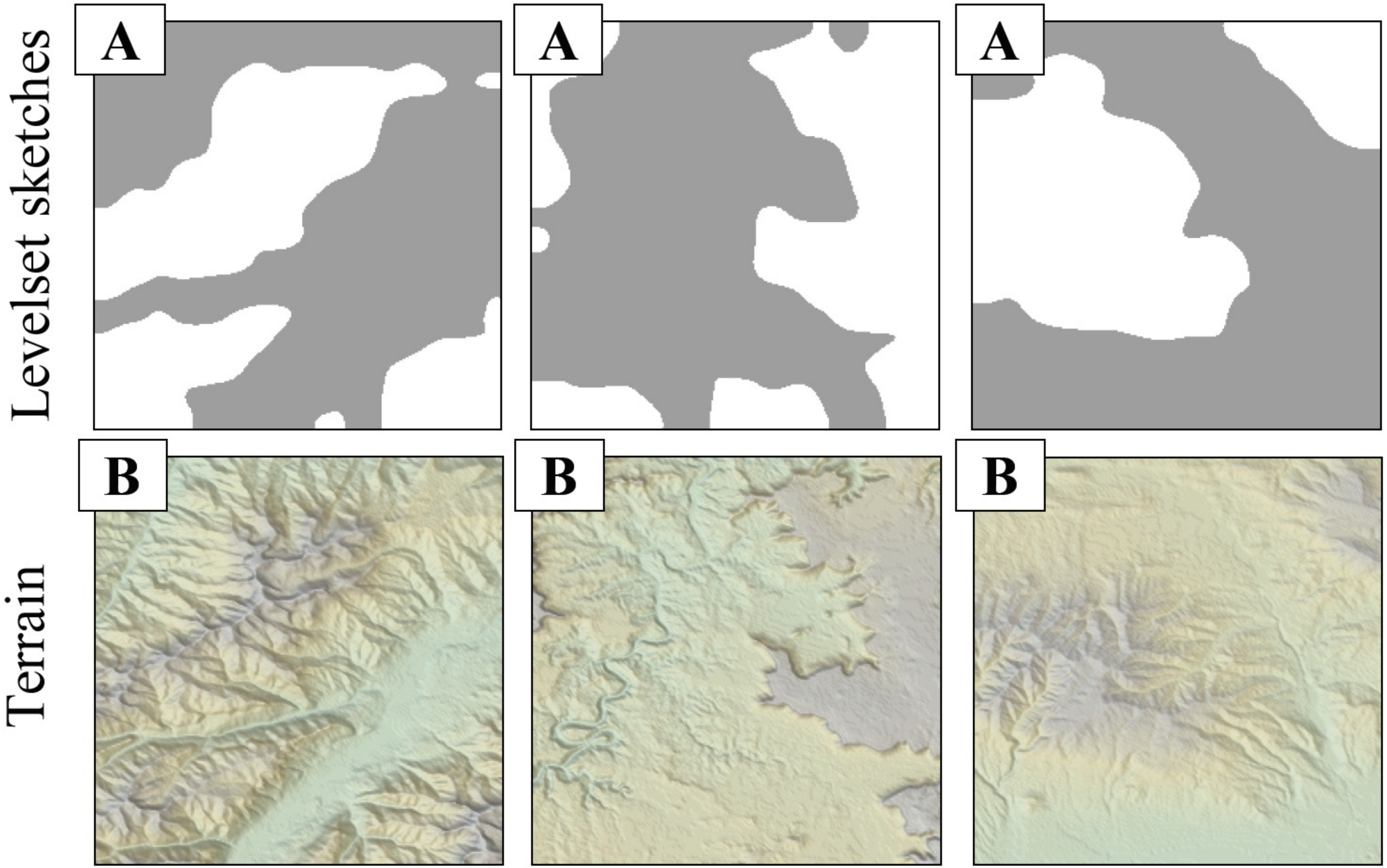


Fig. 9. Levelset examples. The levelset is represented in white.

Training stage – Eraser Synthesizer

- A part of terrain design tool
  - Remove parts of a terrain and infers its completion
- Training by modifying a real-world terrain
  - the addition of a random number of disk with random size to define the missing part
  - $Z\alpha$ as a two-channel image: elevation channel (Z) + erasure channel (α)
  - Erasure part: terrain part is set to 0, α channel is set to 1

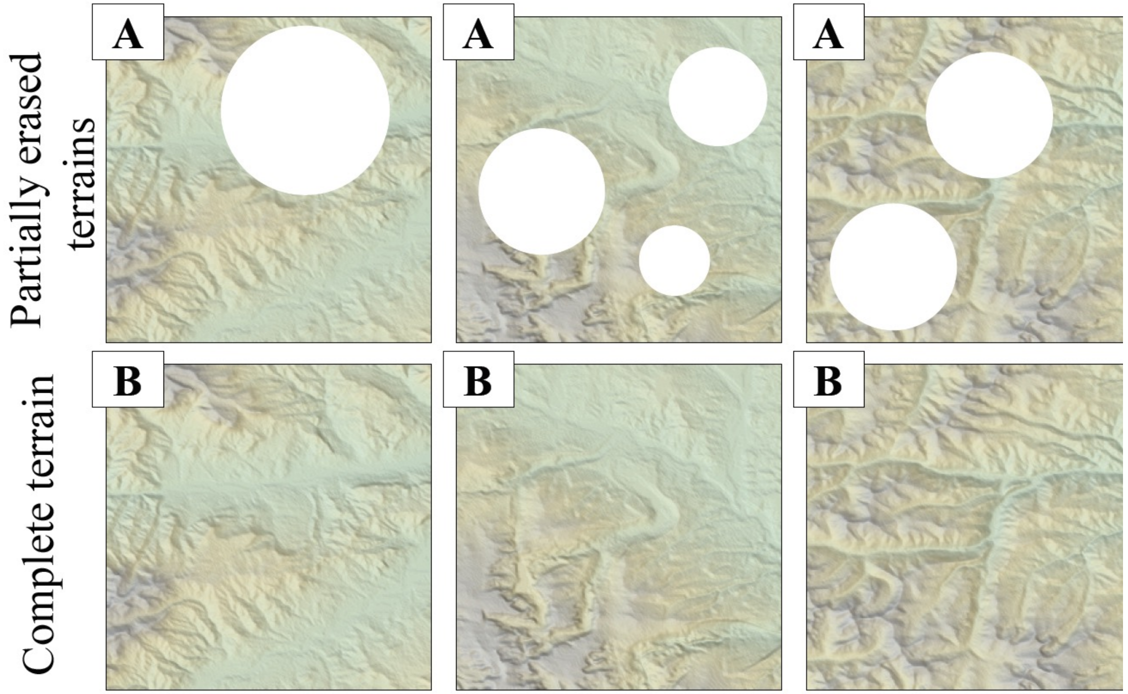


Fig. 11. Eraser synthesizer training example pairs.

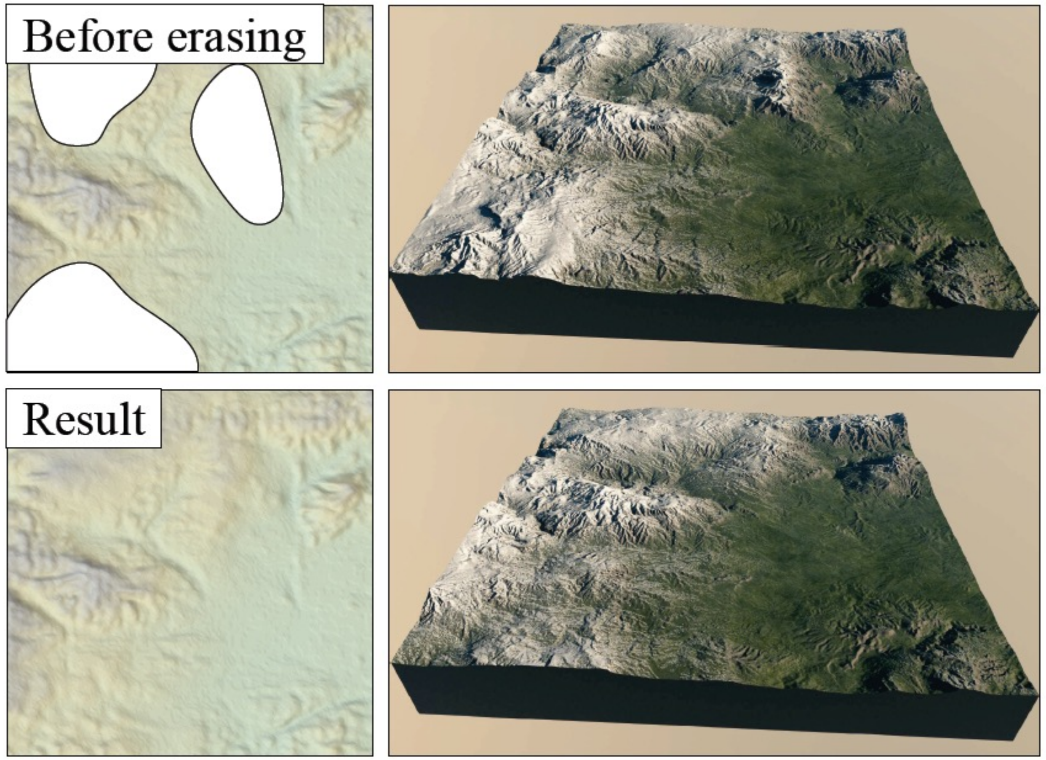


Fig. 14. Example of a terrain automatically generated by the eraser synthesizer tool that fills parts removed by the user.

Training stage – Erosion Synthesizer

- Difficult to find real-world data for a terrain and its corresponding eroded version
- Solution
  - ➢simulating erosion: input real-world terrain A and computing the corresponding data B ($B = e(A)$)

Training stage – Erosion Synthesizer -> simulation algorithm

- Simulate interleaved large-scale hydraulic and thermal erosion
  - ➢Depend on a discrete layered model representing different materials(bedrock, rocks and fine grain sediments)
- Erosion
  - ➢Temperature variations, rainfall
  - ➢Weathering events
    - Water runoff transporting sediments, or fracture of the bedrock into rock-slides
- Stochastically applying a large number of events to the cells of the terrain

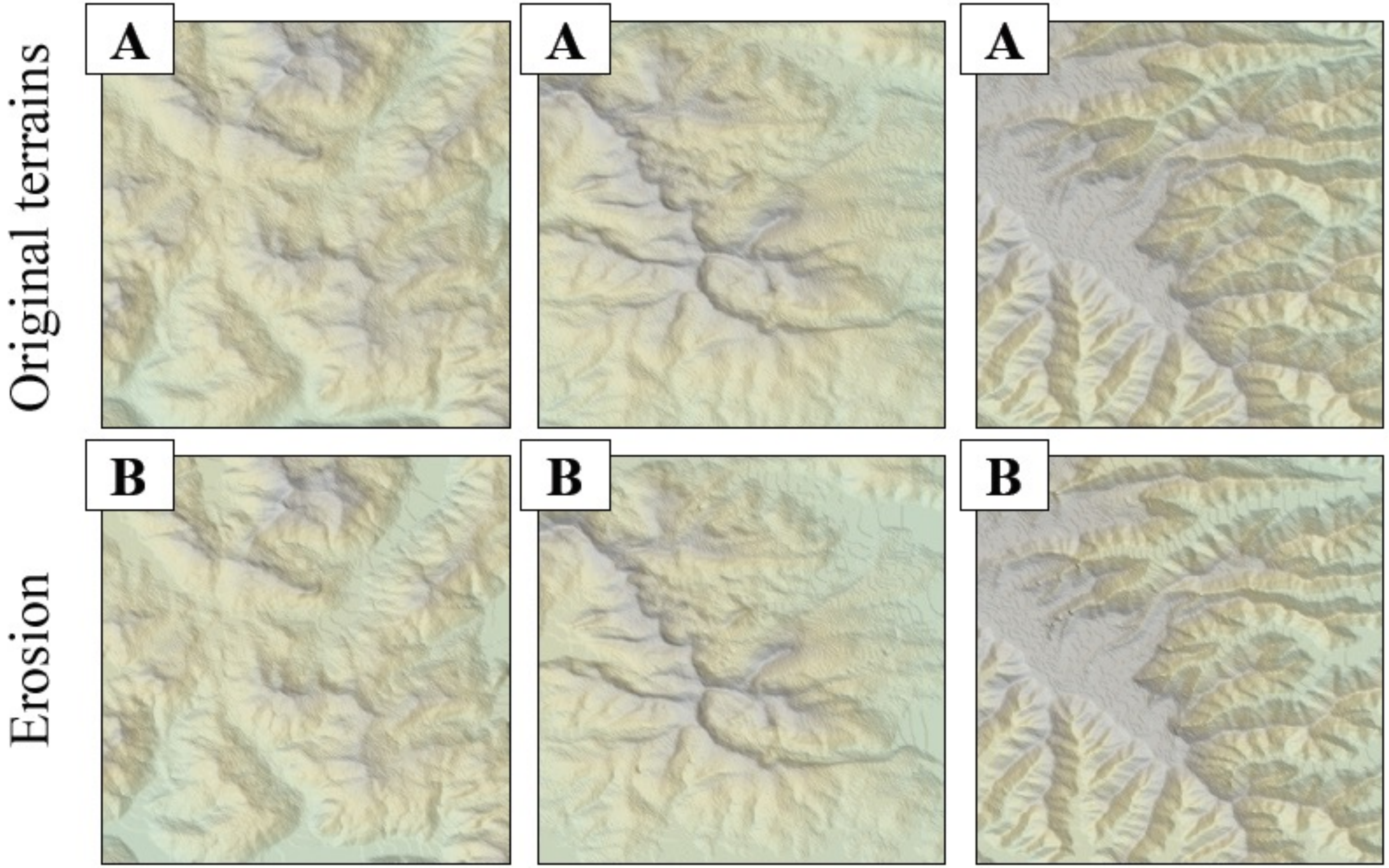
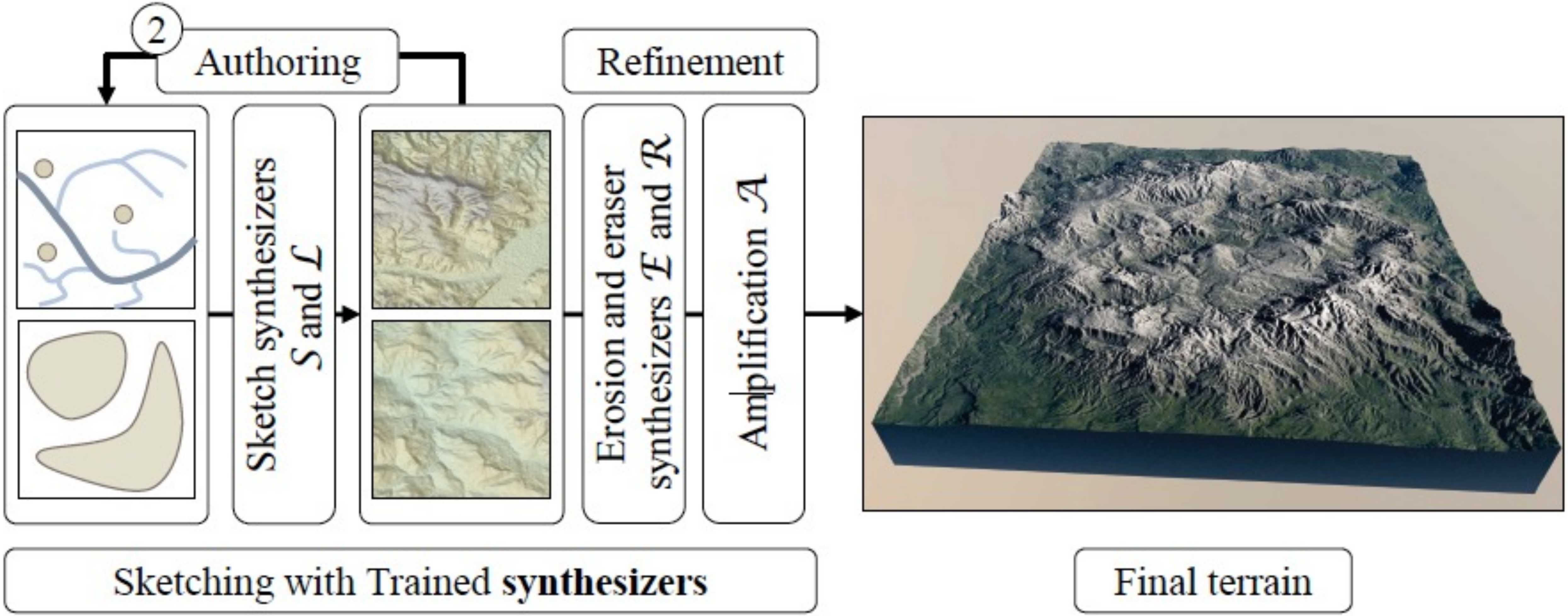
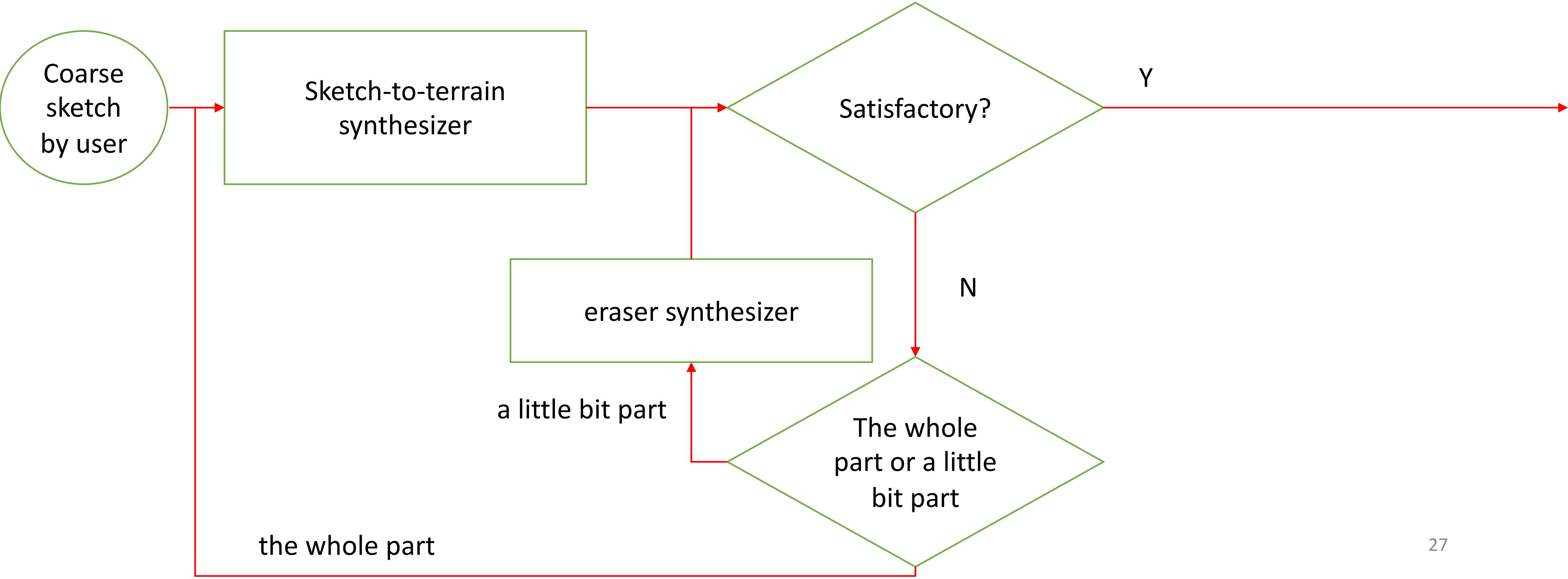


Fig. 12. Erosion synthesizer training pair examples.

1. Info. Of paper
2. Related Work
3. System Pipeline - Training stage
4. **System Pipeline - Inference stage (Authoring)**
5. Results and discussion
6. Limitations, Failure Cases and Conclusion

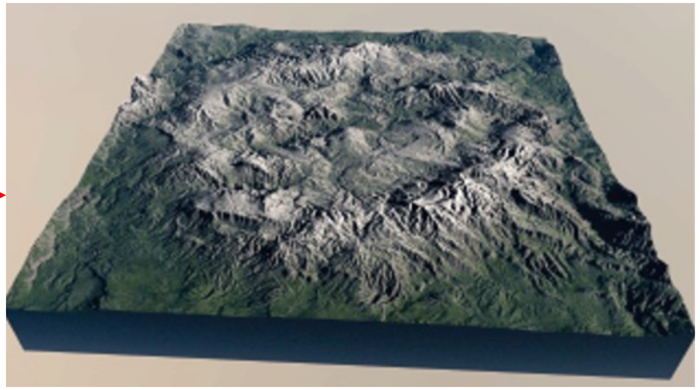

2
Authoring
Refinement
Sketch synthesizers
$\mathcal{S}$ and $\mathcal{L}$
Erosion and eraser
synthesizers $\mathcal{E}$ and $\mathcal{R}$
Amplification $\mathcal{A}$
Sketching with Trained **synthesizers**
Final terrain

(Authoring) Inference stage - 1



(Authoring) Inference stage - 2

Erosion synthesizer → Terrain amplification →



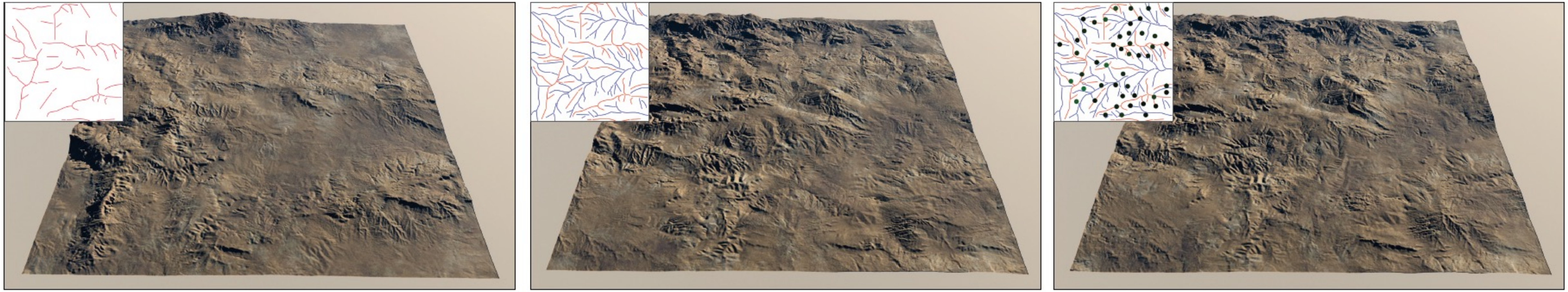


Fig. 10. Example of an interactive authoring session performed by a professional artist: it took him a only a few minutes to design the structure of a large terrain by using ridges (left), adjusting the generated terrain to his intent by incrementally adding rivers (middle), and defining some elevation points (right).

Again: key feature

allows different input types: ridge, river curves, peak, level-set

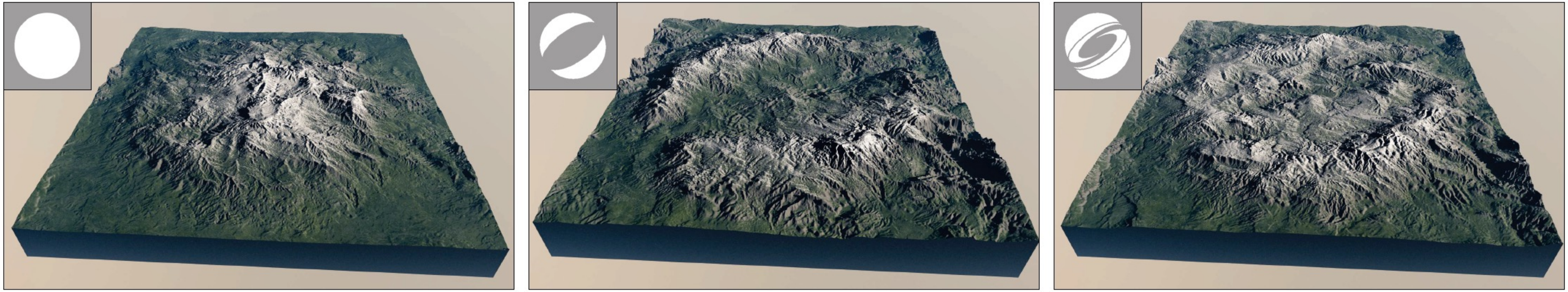


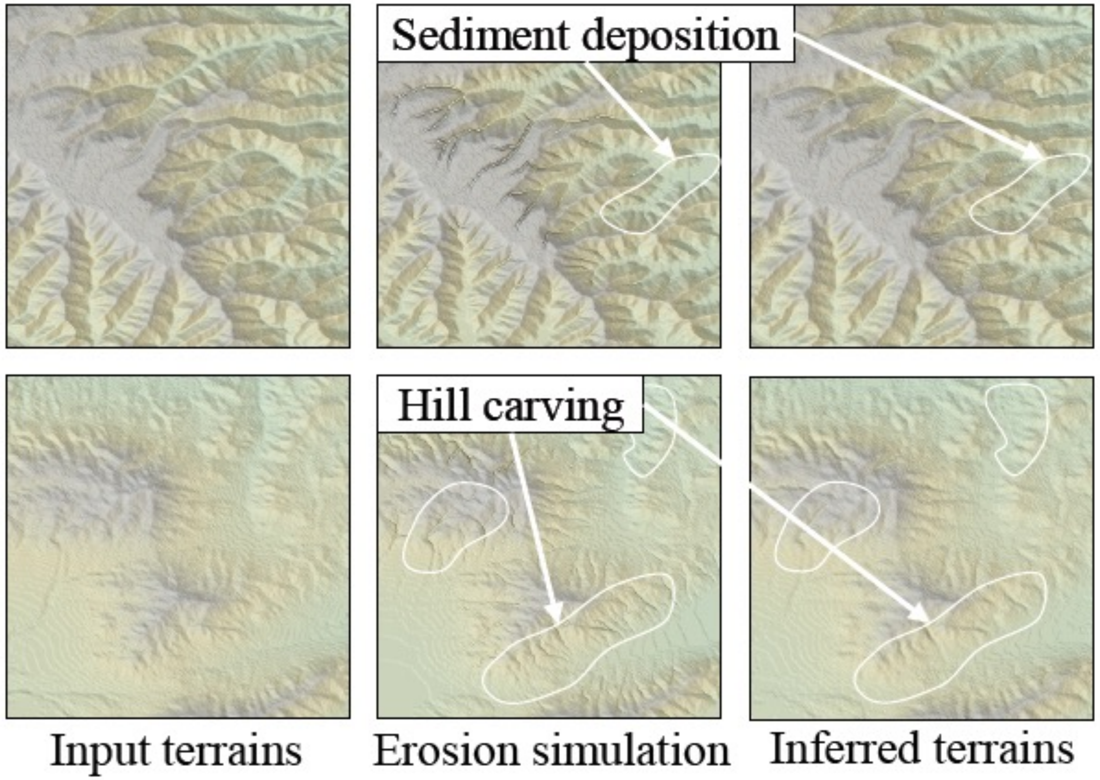
Fig. 13. The iterative sketching can be used to generate complex shapes. Here the user sketches the Siggraph logo by adding a disk, carving a part of the levelset out, and finally adding details. This whole editing sequence is performed using the levelset-to-terrain synthesizer

Terrain Refinement

- Including erosion and amplification
  - ➢ more realism and increase the terrain resolution

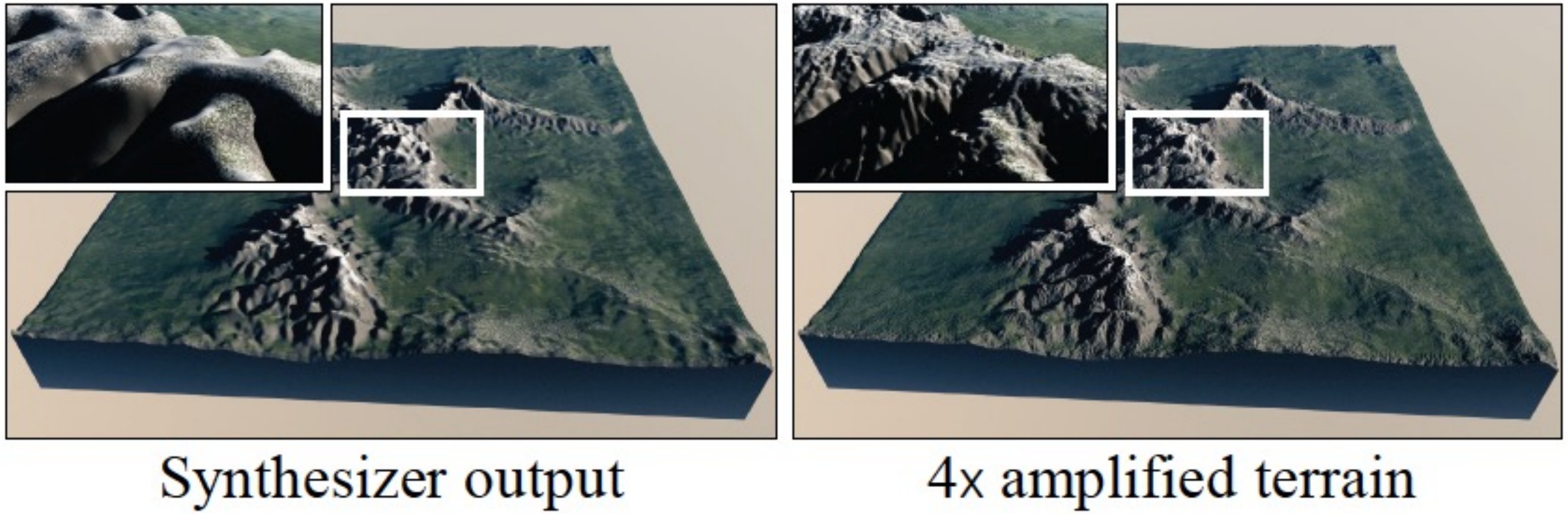
Terrain Refinement - Erosion

- Erosion synthesizer mimics erosion
  - ➢Very fast (25ms vs 40,00ms on a terrain of resolution 256x256)



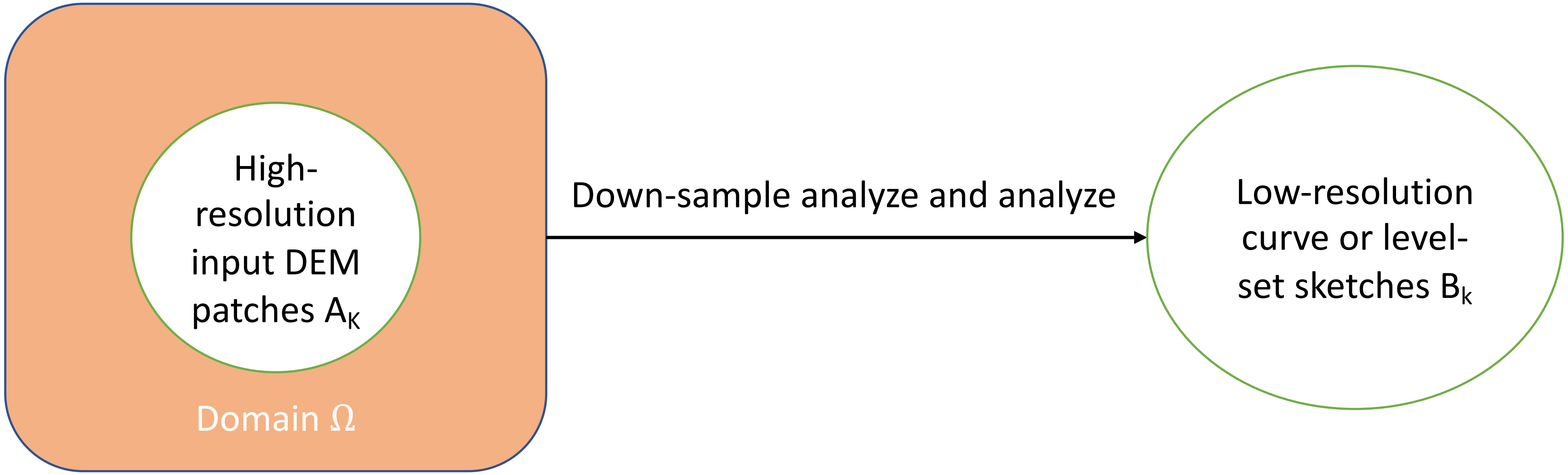
Terrain Refinement – Amplification

- Adding more details on terrain using the <u>patch-based amplification method</u>
  - ➢Builds high and low resolution path dictionaries
  - ➢Decomposes the terrain onto them



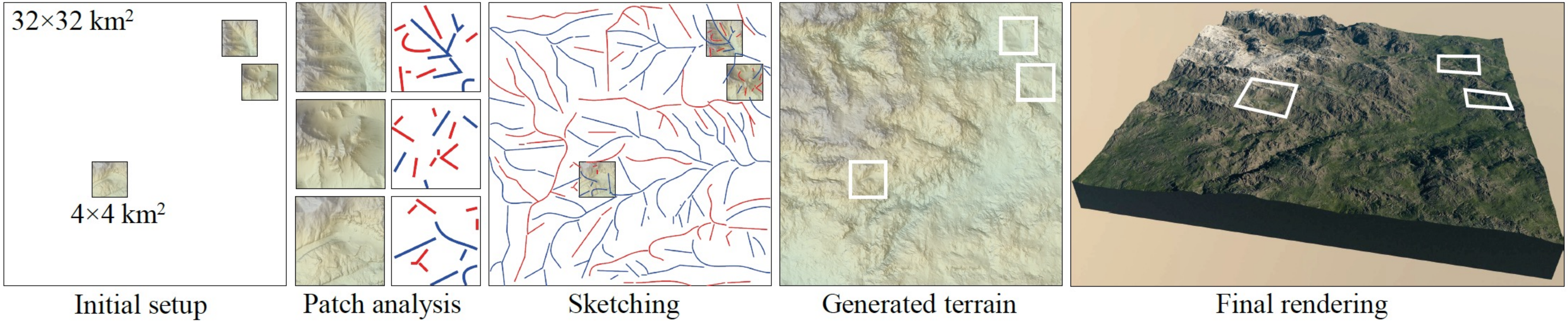
Integration with Large Scale Terrain Modeling ?

- To Enable to generate large scale terrains where some specific regions can author in full detail in a seamless fashion
- User completes the coarse sketch in the remainder of the domain $\Omega - (\cup_k A_k)$ to get a new representation $\mathbb{B}$



Integration with Large Scale Terrain Modeling ?

- Generate the terrain A from $\mathbb{B}$
- Locally smoothly blend it with the patches A_k



1. Info. Of paper
2. Related Work
3. System Pipeline - Training stage
4. System Pipeline - Inference stage (Authoring)
5. **Results and discussion**
6. Limitations, Failure Cases and Conclusion

Results and discussion - Database

(Real-world terrain)DEMs extracted from USGS Earth explorer → Real-world terrain to the coarse sketch (Matlab & C++)

- 35個一平方度的貼片 (精度為一弧秒) from NASA SRTM
  - ➢ 貼圖解像度: 3,600 x 3,600 = 30 x 30 米
  - ➢ 16 bits gray-scale image with a vertical resolution of 1 m

Results and discussion – Size & Training time

Synthesizer	Database creation		Training time
	Size	Time	
Sketch-to-Terrain	525	0:22	6:25
Levelset-to-Terrain	525	0:01	6:24
Eraser	500	0:01	5:48
Erosion	1400	15:13	6:54

Table 1. Timings (in hours) for the learning of terrain synthesizers.

Results and discussion – Comparisons ridges

- [Zhou et al. 2007] example-based based on texture synthesis
  - ➢Generates terrain by combining patches from an input sketch and mountain range style image

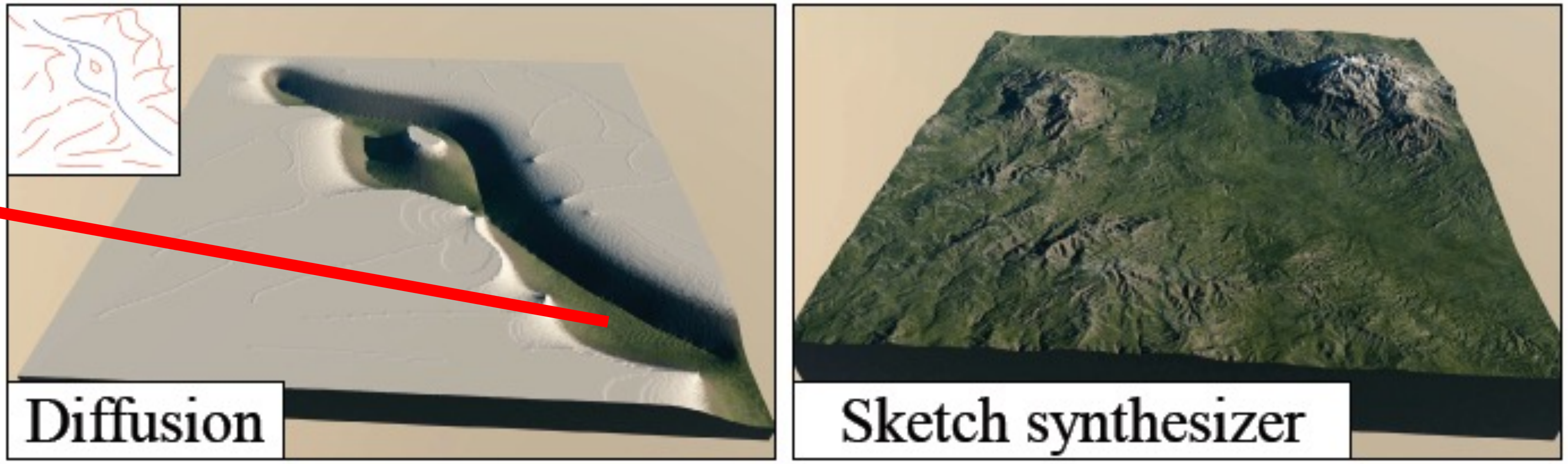
Example-based Our method

Results and discussion – Comparisons ridges and rivers

- [Hnaidi et al. 2010] terrain synthesize by user sketches the ridges and rivers networks
  - ➢Using diffusion to synthesize
  - ➢Limitation: Need to provide more info. to generate the terrain such as: 1. elevation setting 2. strokes to represent the rivers and ridges

Without any derivative restriction

Simple heat diffusion



Results and discussion – Comparisons ridges and rivers

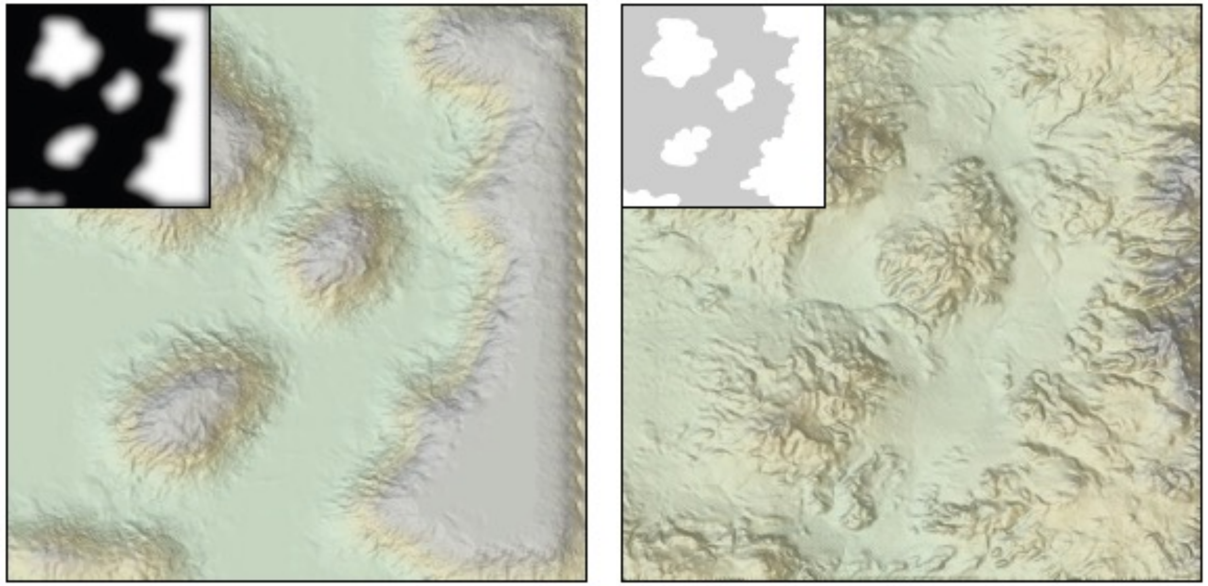
PatchMatch | Gradient PatchMatch | Our method

10s | 15s | 25ms

baseline

Results and discussion – Comparisons levelset

- [Guérin et al. 2016] Sparse method
  - ➢Requires a smooth sketch as input
  - ➢Result: unrealistic result when the input have large scale feature



Sparse method Our method

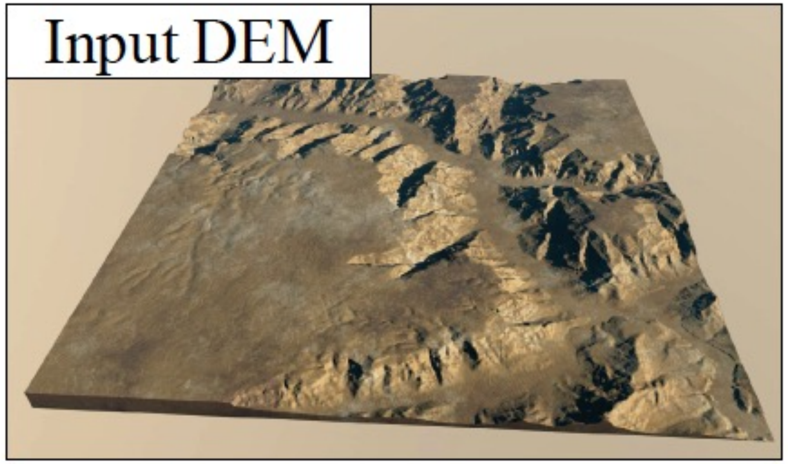
Results and discussion – Comparisons erosion

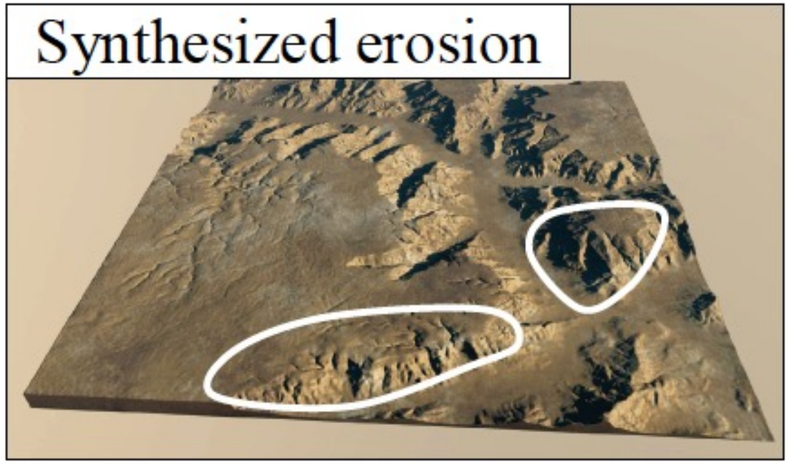
- 4,000x fast than erosion simulations
  - May contain some geologically incorrect features

Erosion simulation
741.0s

VS

cGAN-based method
0.7s



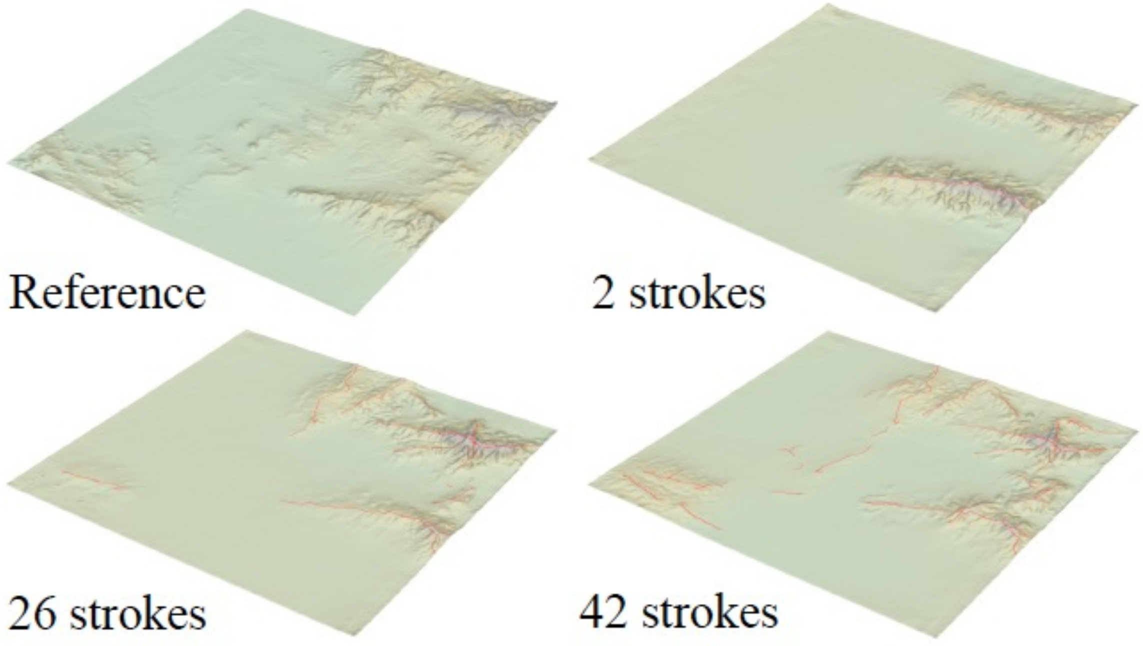


Statistics for interactive authoring: terrain size and processing time(in ms)

Process	Terrain Size	Time (ms)
Synthesizers $\mathcal{L}$ and $\mathcal{S}$	256×256	25
	512×512	55
	1024×1024	190
Optional erosion $\mathcal{E}$ or eraser $\mathcal{R}$	1024×1024	190
Interactive feedback	512×512	310
Optional amplification ($\times 4$)	$256^2 \rightarrow 1024^2$	800
	$512^2 \rightarrow 2048^2$	3250

Performance, User Control, and Experience

- Performance: interactivity
- High usability: all participants(including novice and expert) were able to express their intent



A qualitative user-study (using Likert scale)

5 users:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
Does the generated terrain follow the sketch?	0	0	0		5
Is the system reactive?	0	0	0		5
Is it easy to express ones intent?	0	1	0		4

1. Info. Of paper
2. Related Work
3. System Pipeline - Training stage
4. System Pipeline - Inference stage (Authoring)
5. Results and discussion
6. **Limitations, Failure Cases and Conclusion**

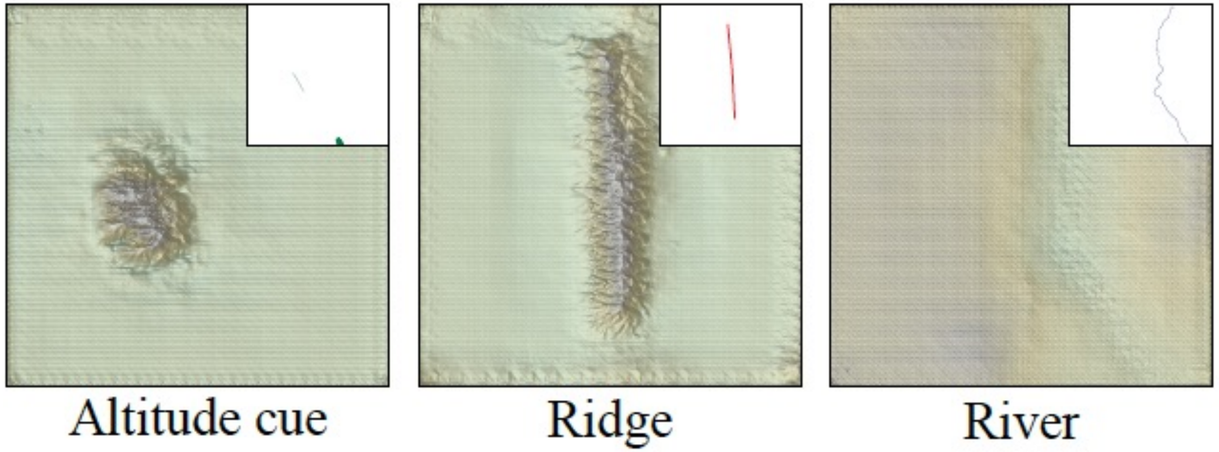
Limitations and Failure Cases

- Limitations
  1. Need to retrain the synthesizer if someone wants to use/add a different kind of sketch
  2. User must learn to draw a certain type of sketch by the synthesizer
  3. Leveset and curve cannot be used simultaneously

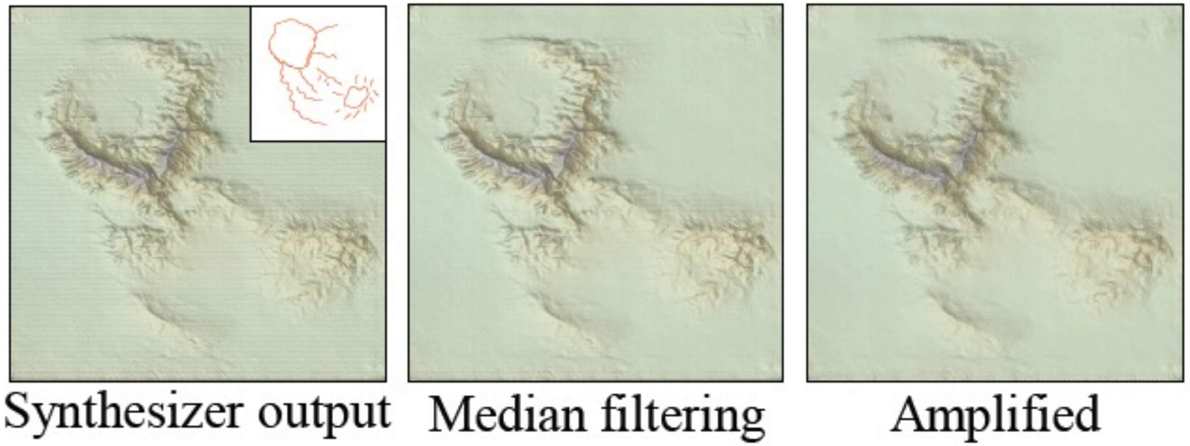
Limitations and Failure Cases

- Failure Cases
    1. A strong repetition effect will appear if the sketch is spare
    2. When no sketch cues is available (the terrain is flat due to a lack of input cues), the synthesized terrain may exhibit some regular pattern
- Improve
    1. Adding more strokes
    2. Post-processing by applying a 5 x 5 median filter

repetition effect



regular pattern



CONCLUSION

- Propose a novel framework for modelling terrains from input sketches
- The heart of the framework:
  - ➢learning the relationship: 地形特徵 ⟷ 數值地形高程資料
- Efficient
  - ➢allowing interactive feedback to the designer
- Users can create large scale realistic models quickly and easily

Future Work

- Bind a procedural model to the system
  - ➢E.x.: the procedural primitive-based terrain representation

- learn the parameters to get a complete inverse procedural modeling system

- To model terrains with different material layers