AutoBiomes

Procedural Generation of Multi-Biome Landscapes

Roland Fischer, Philipp Dittmann, René Weller, Gabriel Zachmann

University of Bremen, Germany

rfischer@cs.uni-bremen.de

CGI, 20-23 October 2020, Geneva

Motivation

The Good Dinosaur [Disney/Pixar]

Motivation

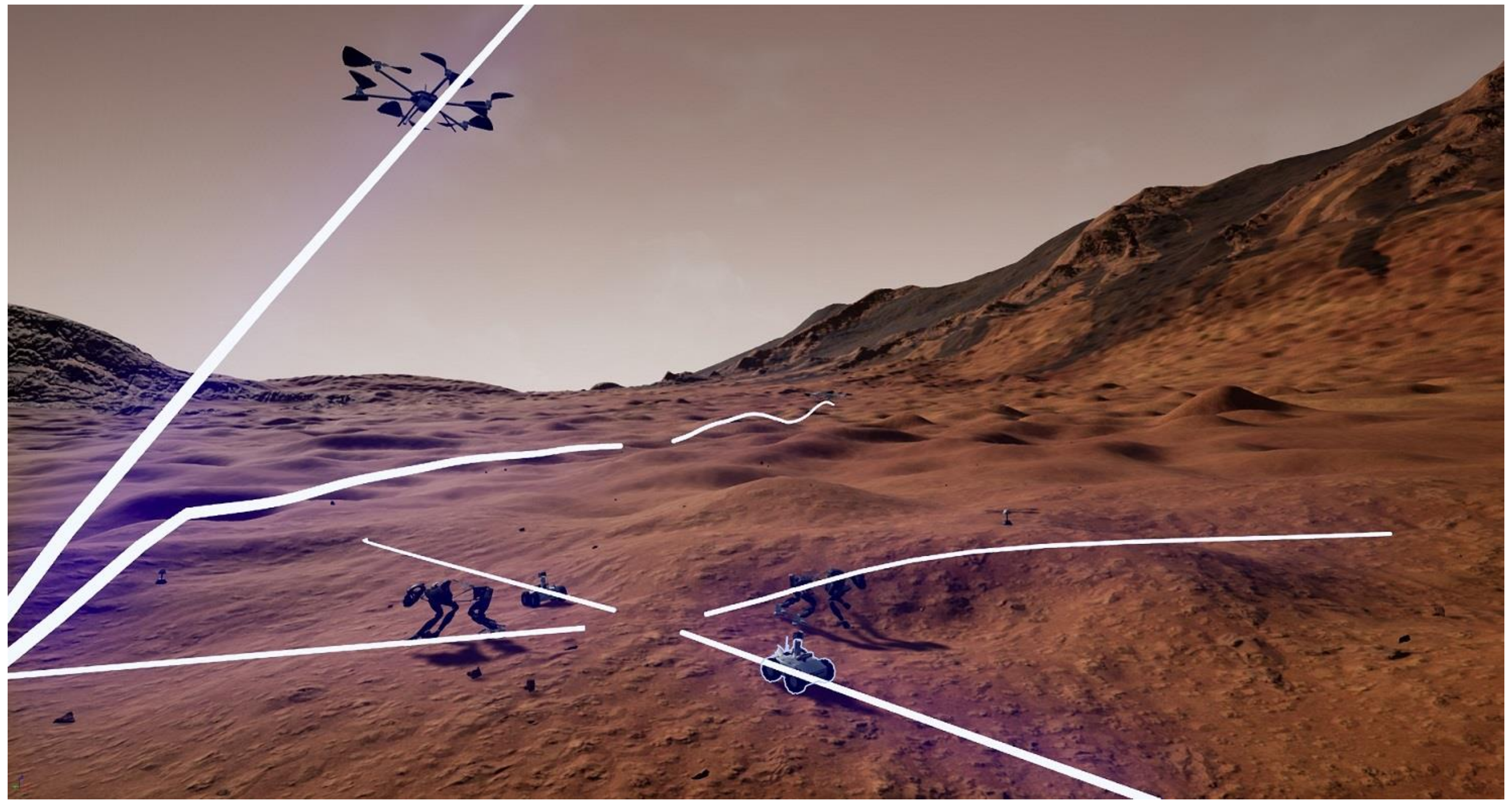
The Good Dinosaur [Disney/Pixar]

Ghost Recon: Wildlands [Ubisoft]

Motivation

The Good Dinosaur [Disney/Pixar]

Ghost Recon: Wildlands [Ubisoft]



VaMEx-VTB [University of Bremen]

Motivation

- Huge landscapes as combination of different biomes

Minecraft

Motivation

- Huge landscapes as combination of different biomes
- Populating the terrain with objects


Minecraft

Motivation

- Huge landscapes as combination of different biomes
- Populating the terrain with objects
- Manual creation is not an option

Minecraft

Motivation

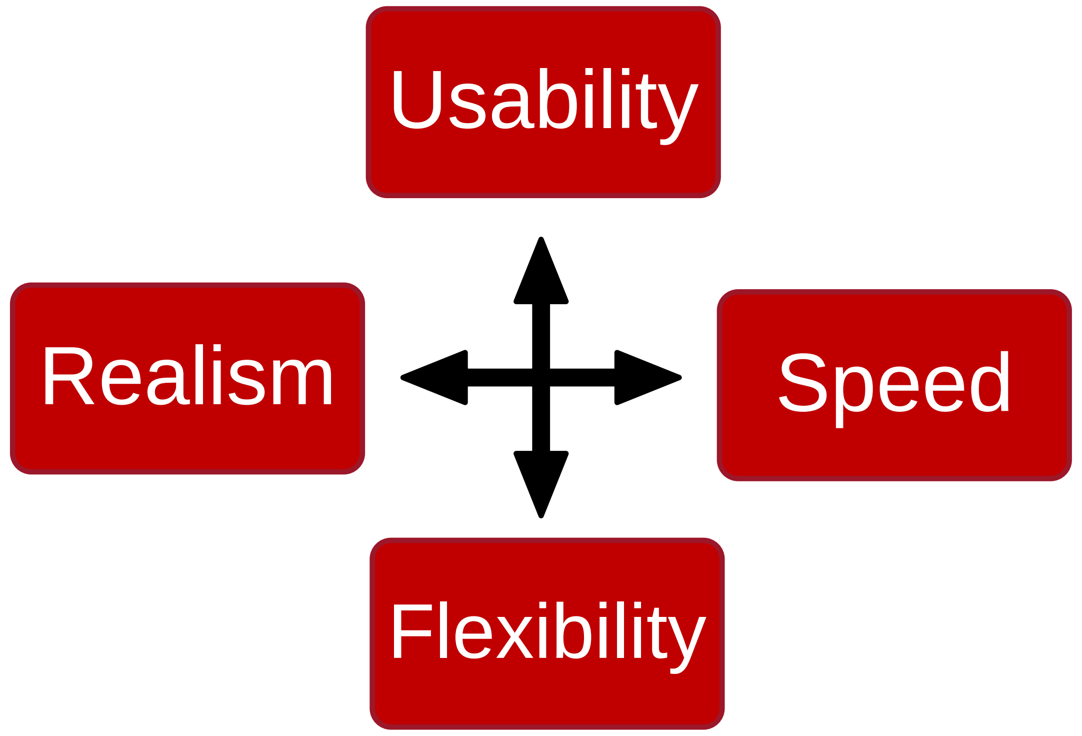
- Huge landscapes as combination of different biomes
- Populating the terrain with objects
- Manual creation is not an option
- Procedural terrain generation (PTG)

Minecraft

Motivation

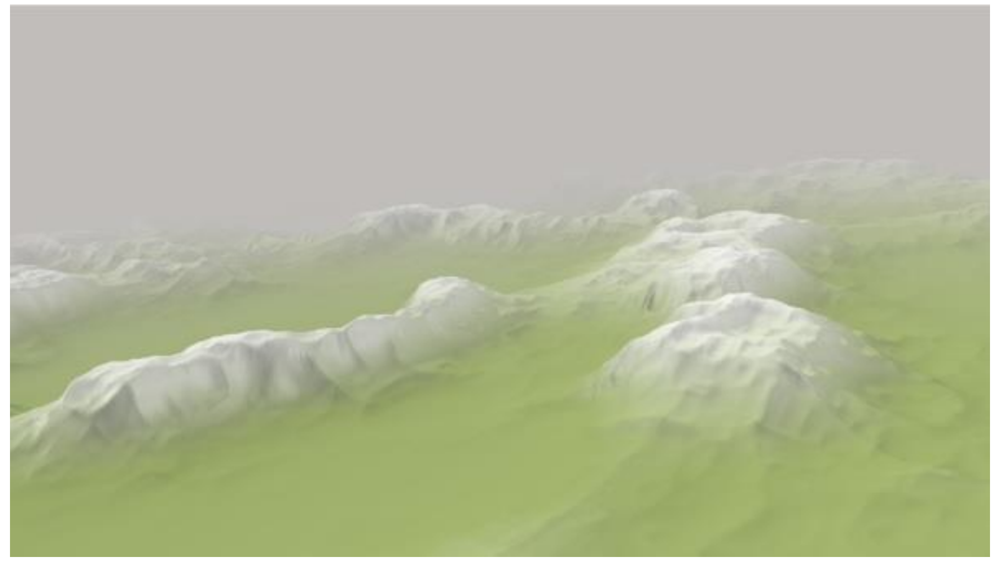
- Huge landscapes as combination of different biomes
- Populating the terrain with objects
- Manual creation is not an option
- Procedural terrain generation (PTG)
  - Much researched, still open challenges

Minecraft



Related Work

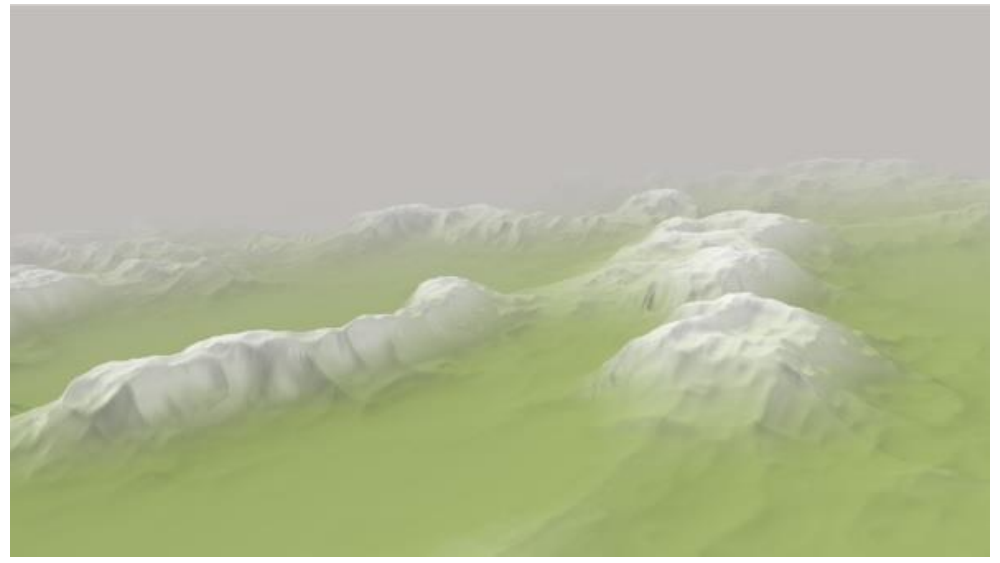
- Main PTG approaches:
  - Synthetic, e.g. noise
    - Fast, unintuitive, hard to get realistic results [Thorimbert18]



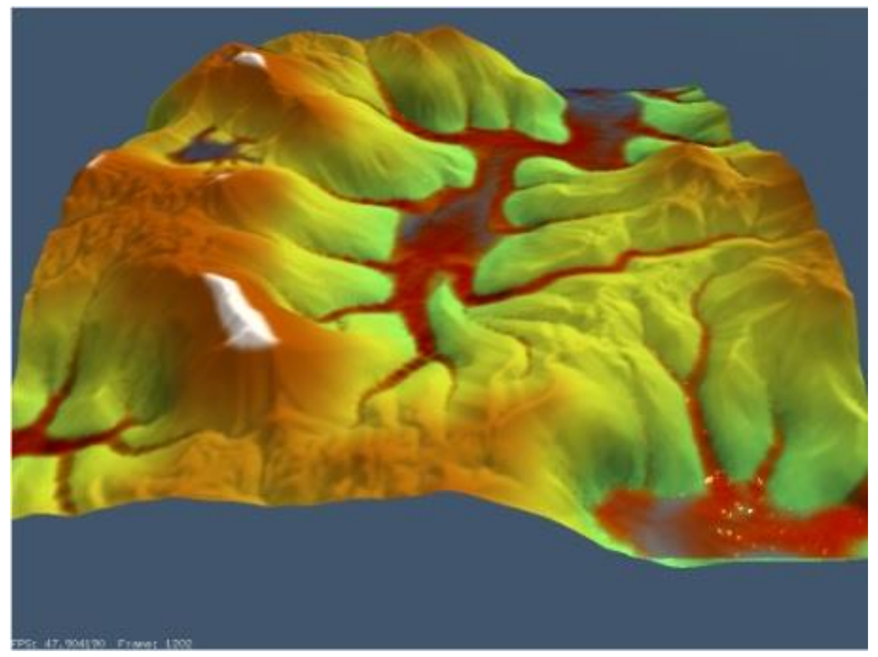
[Thorimbert18]

Related Work

- Main PTG approaches:
  - Synthetic, e.g. noise
    - Fast, unintuitive, hard to get realistic results [Thorimbert18]
  - Physics-based, e.g. erosion, fluid simulation
    - Complex, realistic results, slow [Stam03, Jákó11, Ihmsen14]



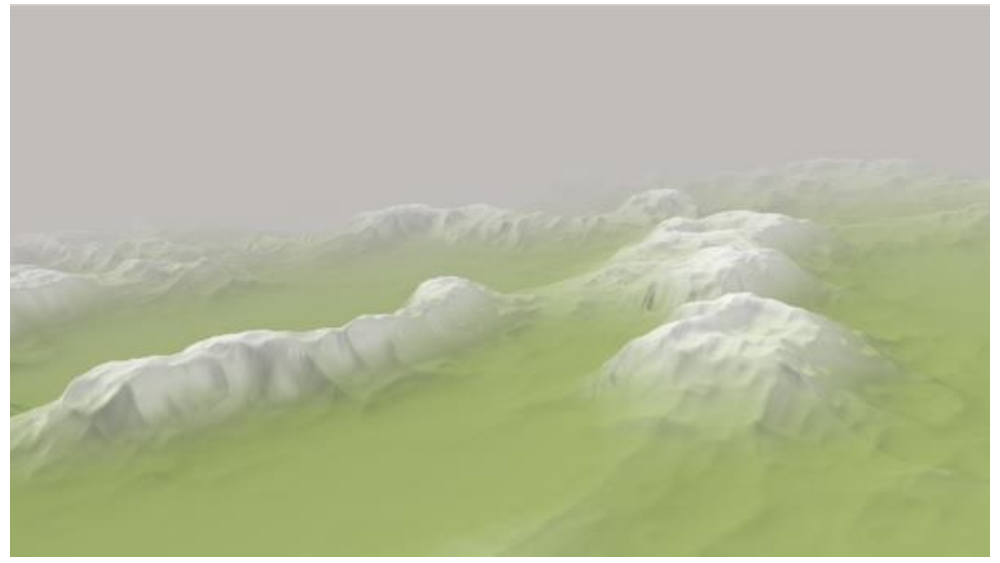
[Thorimbert18]



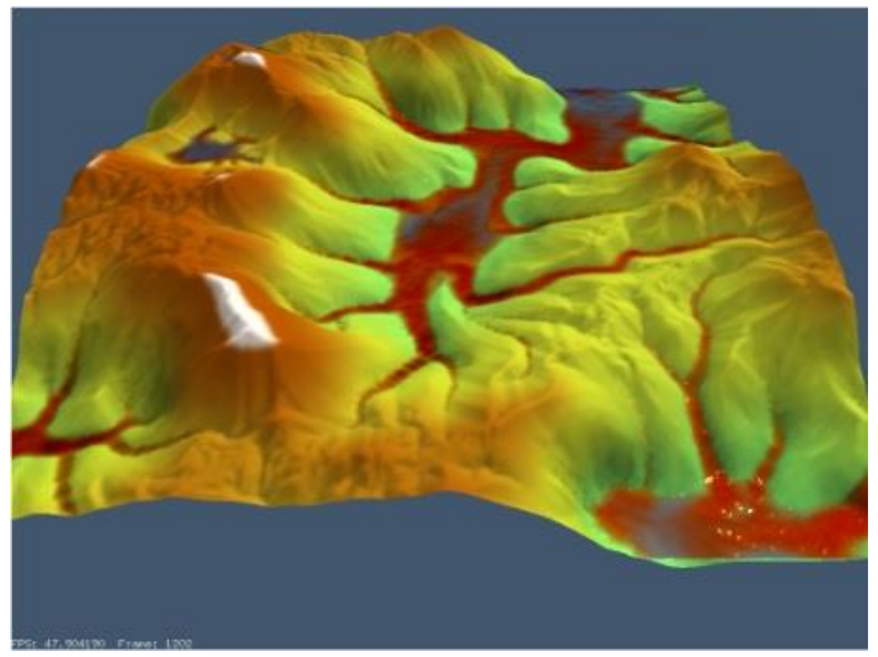
[Jákó11]

Related Work

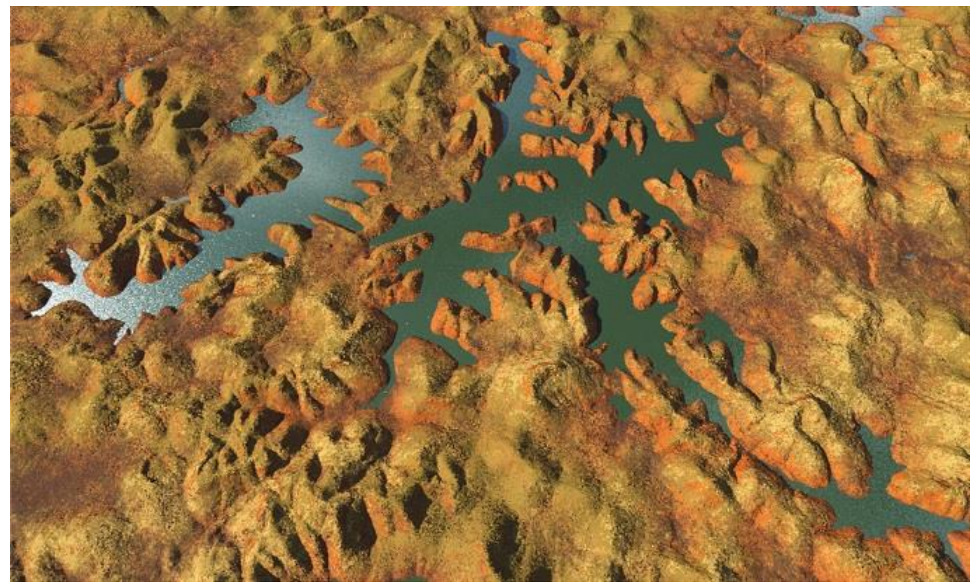
- Main PTG approaches:
  - Synthetic, e.g. noise
    - Fast, unintuitive, hard to get realistic results [Thorimbert18]
  - Physics-based, e.g. erosion, fluid simulation
    - Complex, realistic results, slow [Stam03, Jákó11, Ihmsen14]
  - Example-based, e.g. image synthesis, DEMs, neural networks
    - Realistic, good usability, inflexible [Zhou07, Beckham17, Wulff-Jensen18]



[Thorimbert18]



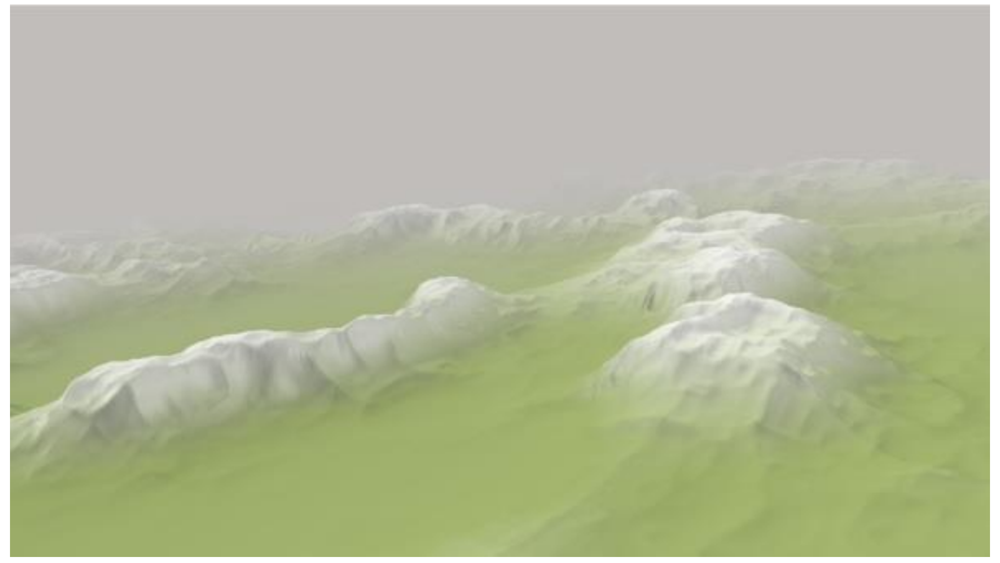
[Jákó11]

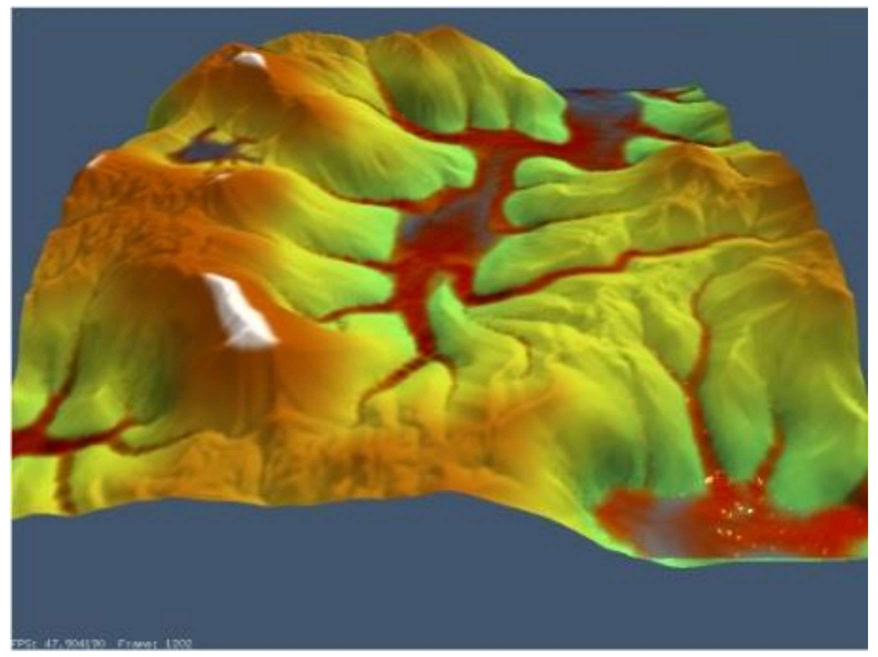


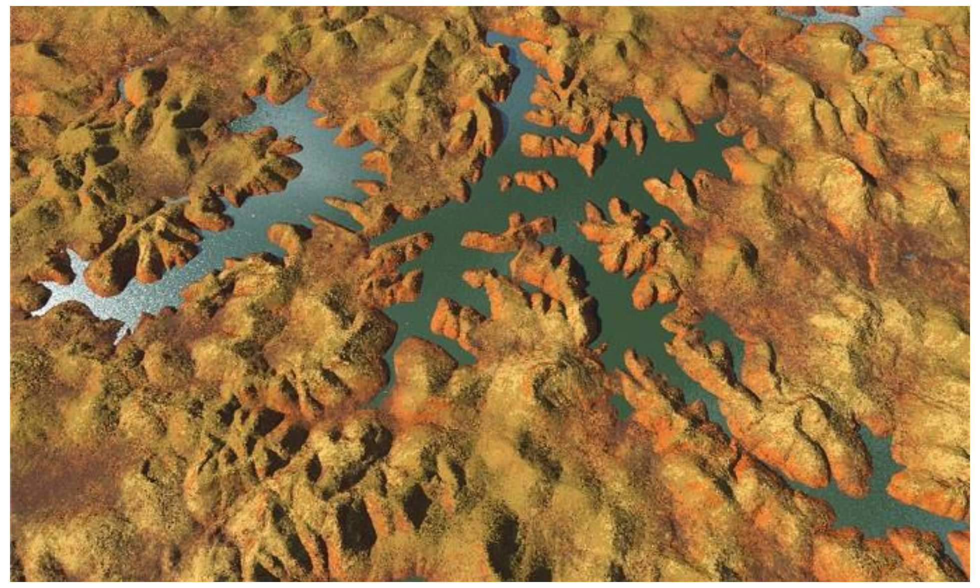
[Zhou07]

Related Work

- Main PTG approaches:
  - Synthetic, e.g. noise
    - Fast, unintuitive, hard to get realistic results [Thorimbert18]
  - Physics-based, e.g. erosion, fluid simulation
    - Complex, realistic results, slow [Stam03, Jákó11, Ihmsen14]
  - Example-based, e.g. image synthesis, DEMs, neural networks
    - Realistic, good usability, inflexible [Zhou07, Beckham17, Wulff-Jensen18]
- Also valid for commercial tools (e.g. World Creator, World Machine, Terragen)


[Thorimbert18]


[Jákó11]


[Zhou07]

Our Contributions

- Novel PTG system combining advantages of 3 approaches

Our Contributions

- Novel PTG system combining advantages of 3 approaches
- Effective generation of vast, plausible-looking terrains

Our Contributions

- Novel PTG system combining advantages of 3 approaches
- Effective generation of vast, plausible-looking terrains
- Multi-biome landscapes

Our Contributions

- Novel PTG system combining advantages of 3 approaches
- Effective generation of vast, plausible-looking terrains
- Multi-biome landscapes
- Dense, complex asset distribution

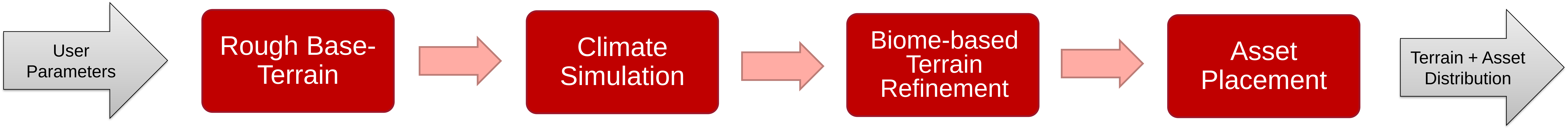
Our Contributions

- Novel PTG system combining advantages of 3 approaches
- Effective generation of vast, plausible-looking terrains
- Multi-biome landscapes
- Dense, complex asset distribution
- Easy-to-use, iterative workflow

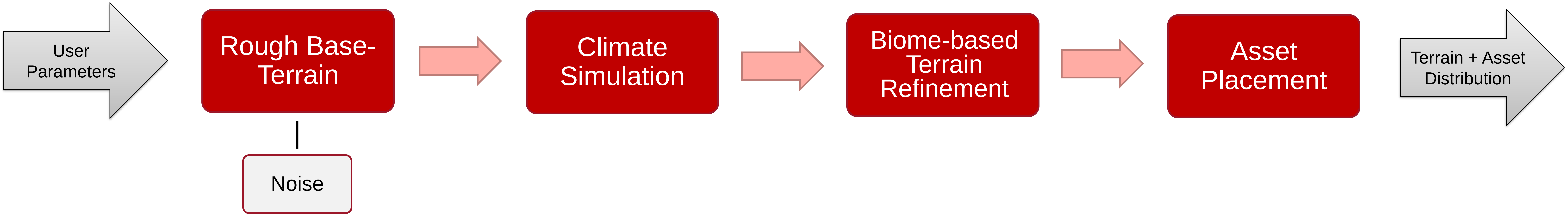
Our Contributions

- Novel PTG system combining advantages of 3 approaches
- Effective generation of vast, plausible-looking terrains
- Multi-biome landscapes
- Dense, complex asset distribution
- Easy-to-use, iterative workflow
- Unreal Engine 4 integration

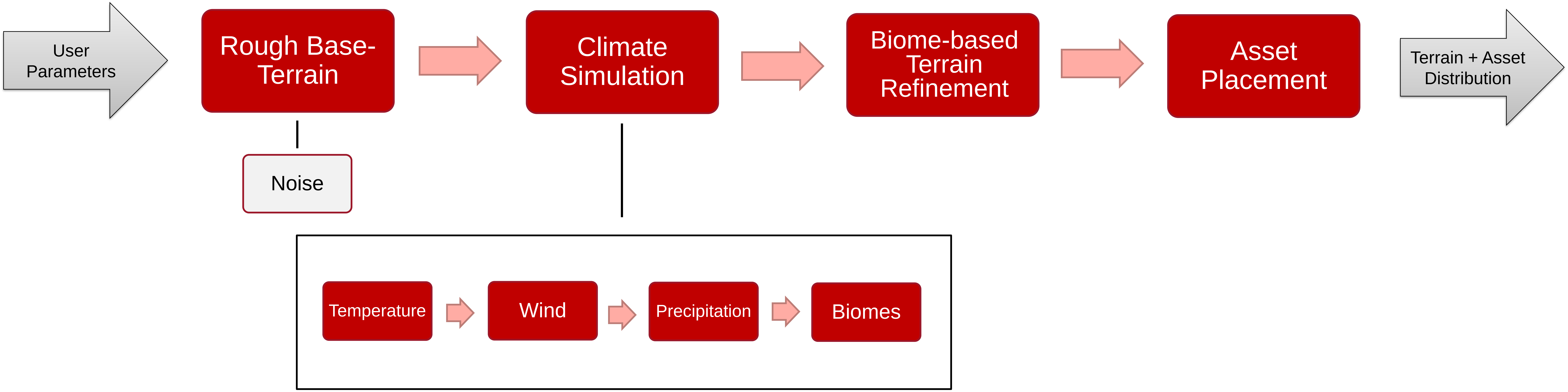
Our PTG Pipeline



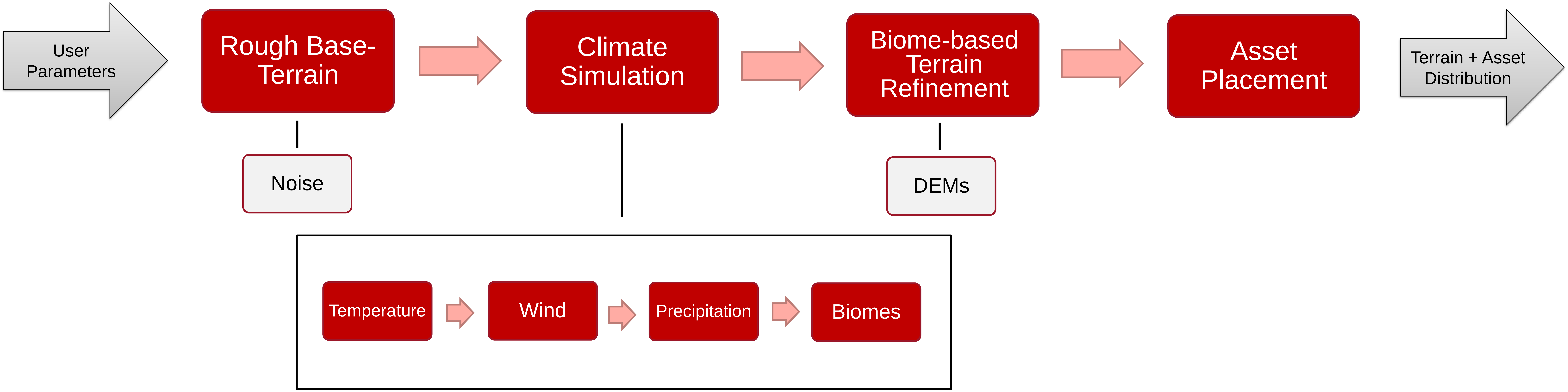
Our PTG Pipeline



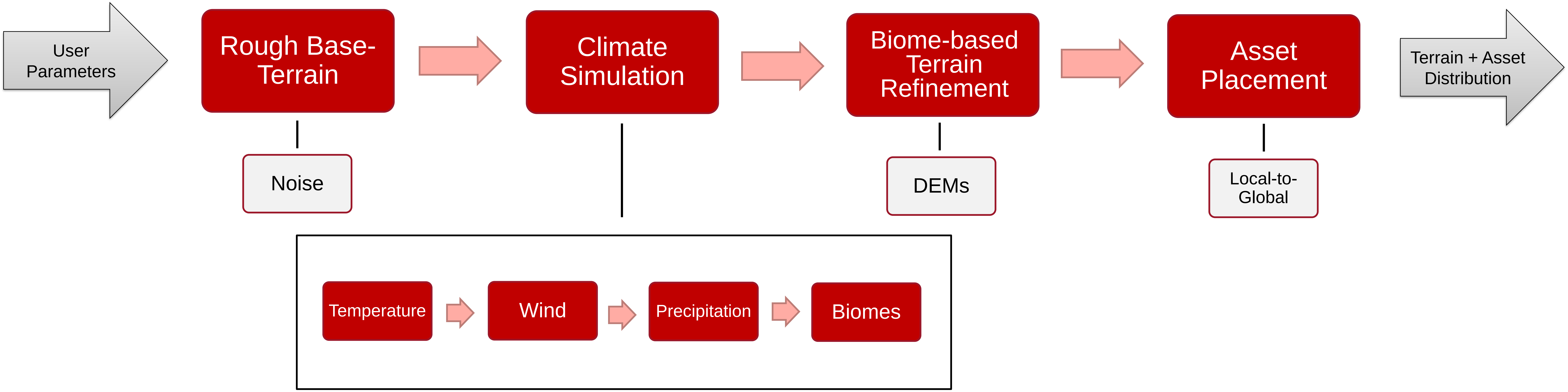
Our PTG Pipeline



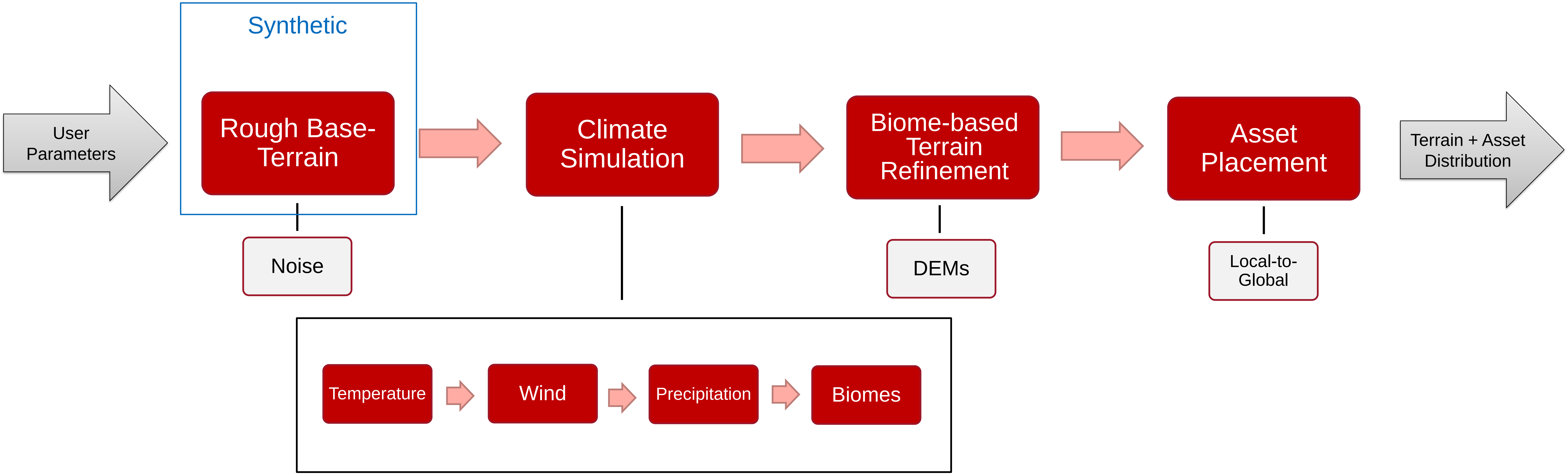
Our PTG Pipeline



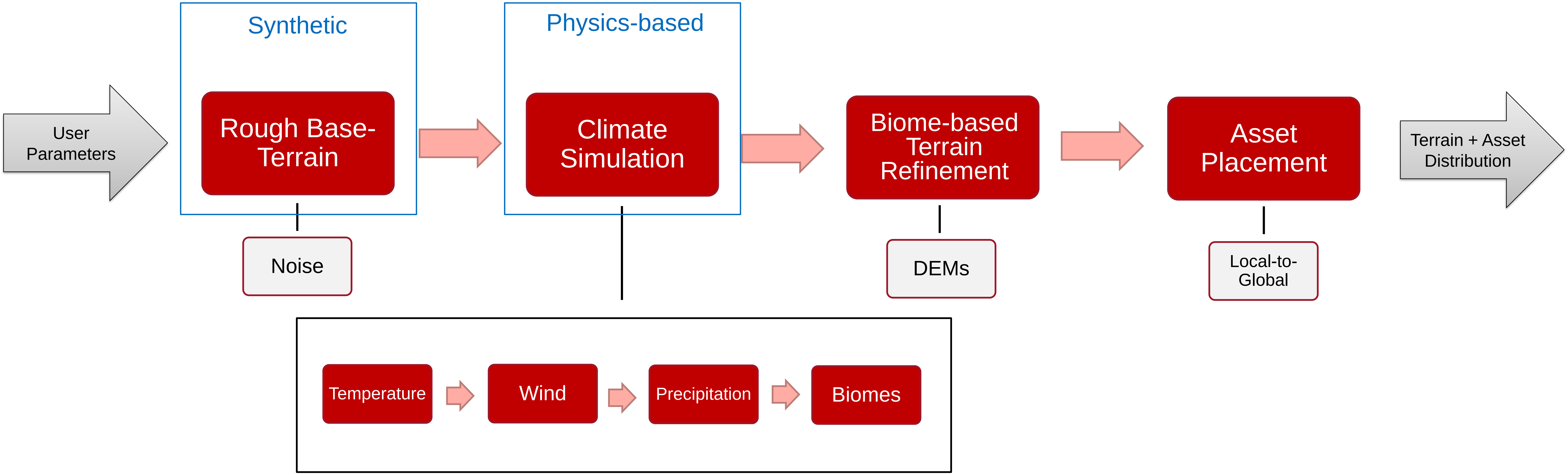
Our PTG Pipeline



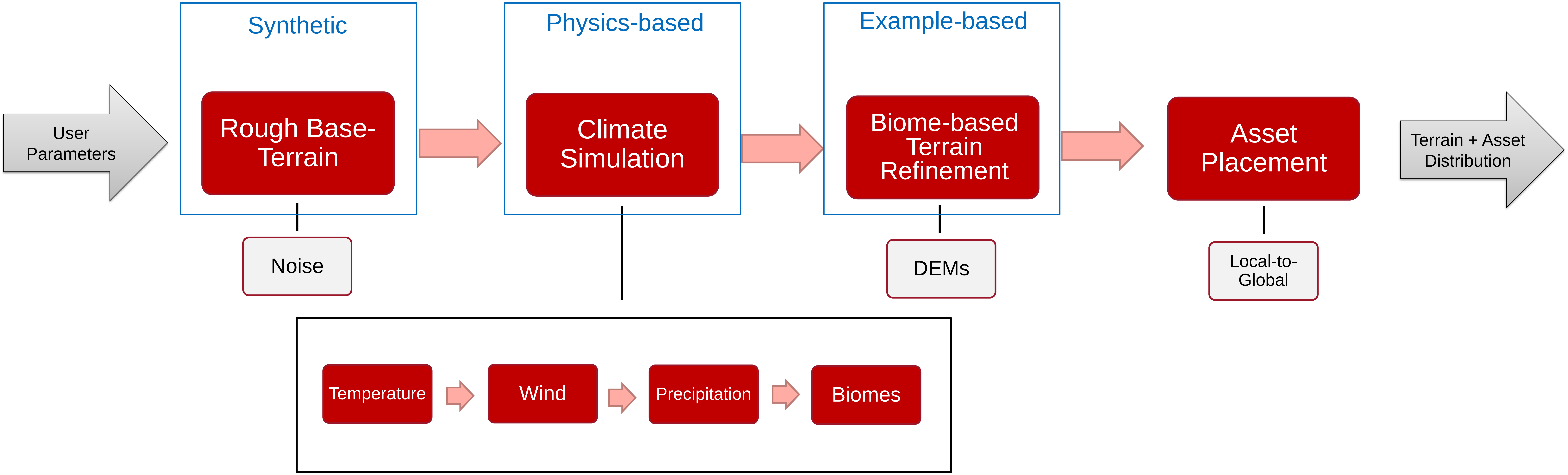
Our PTG Pipeline



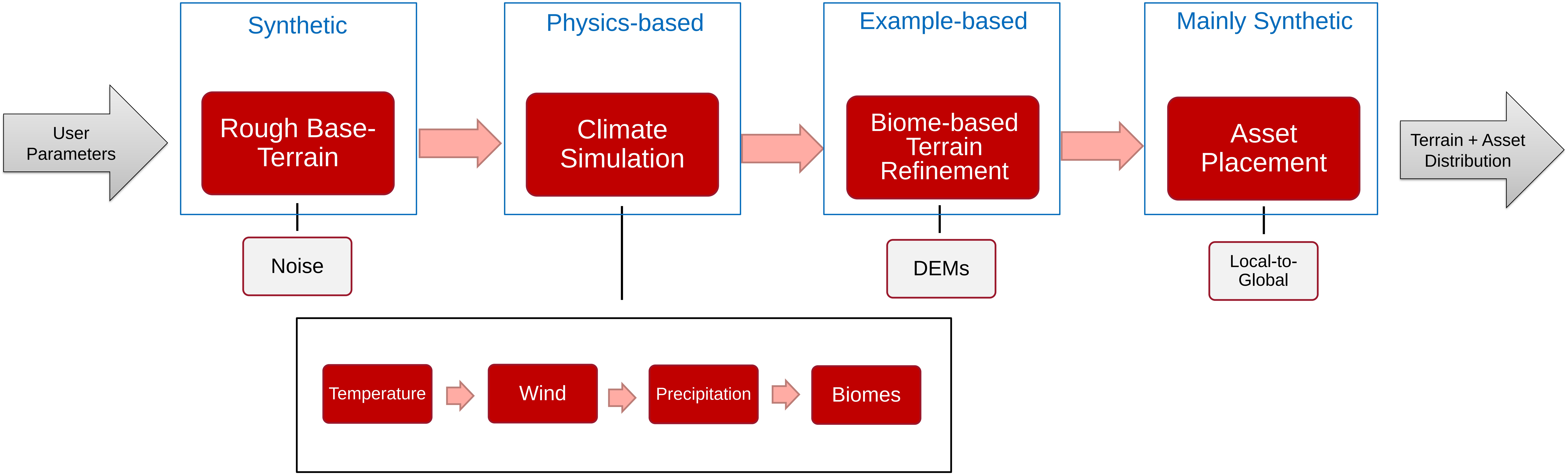
Our PTG Pipeline



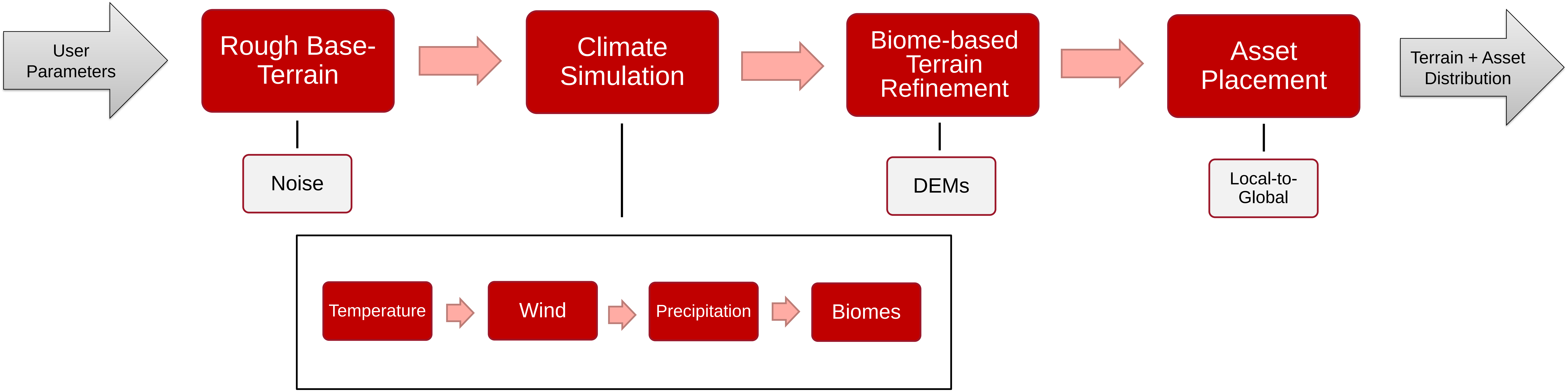
Our PTG Pipeline



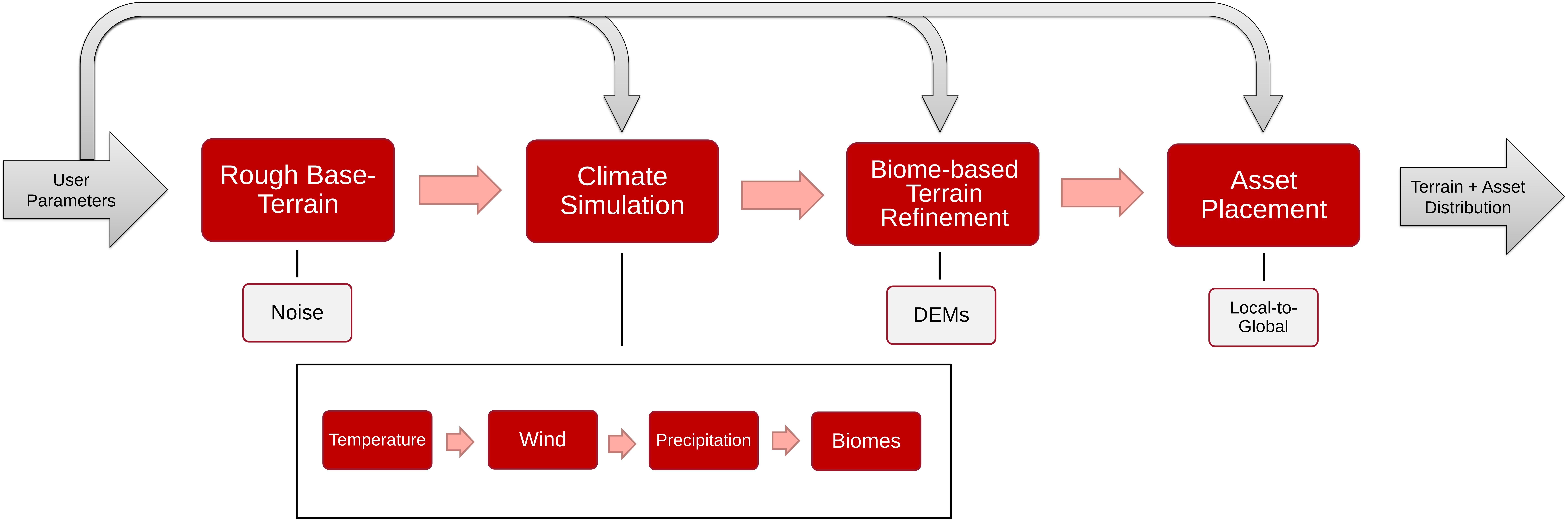
Our PTG Pipeline



Our PTG Pipeline

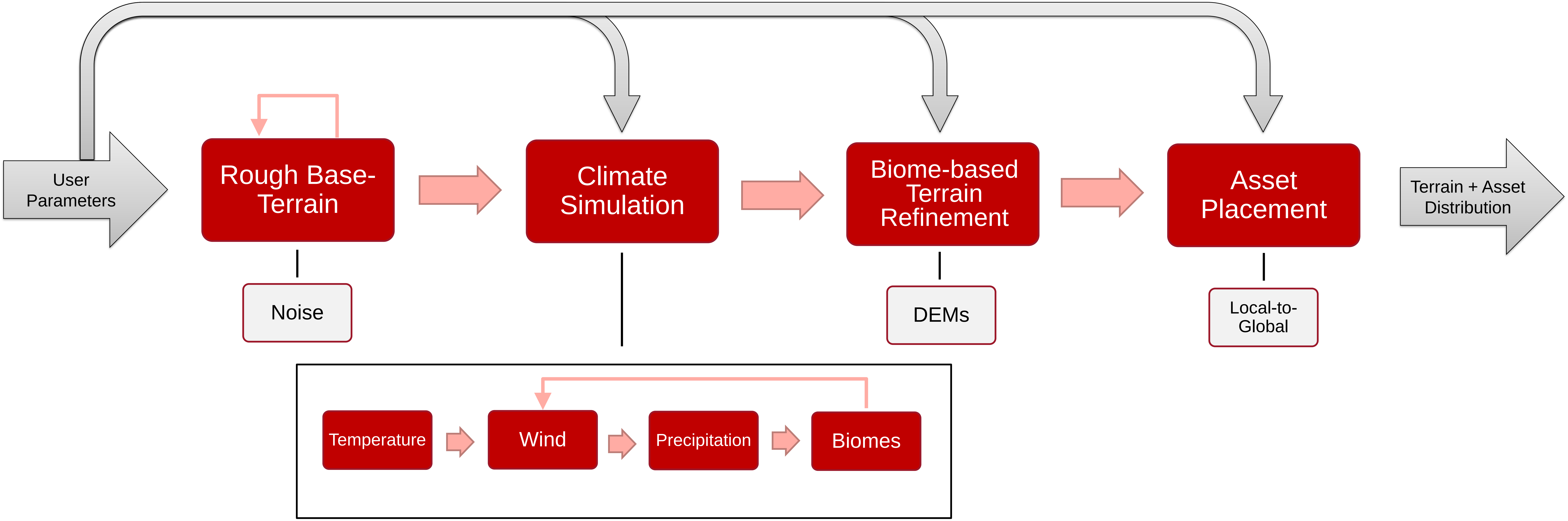


Our PTG Pipeline



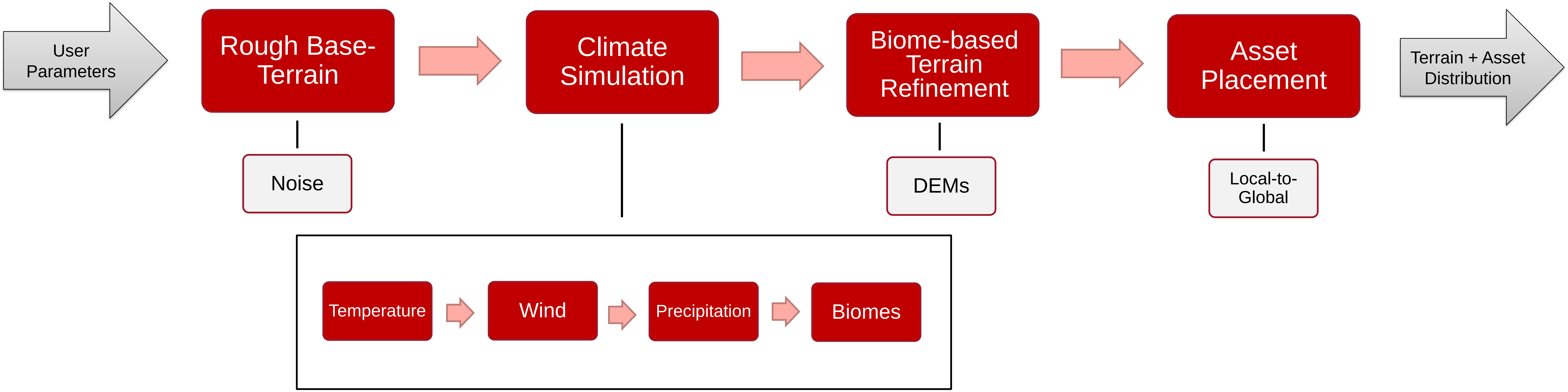
- Each step is customizable and repeatable

Our PTG Pipeline



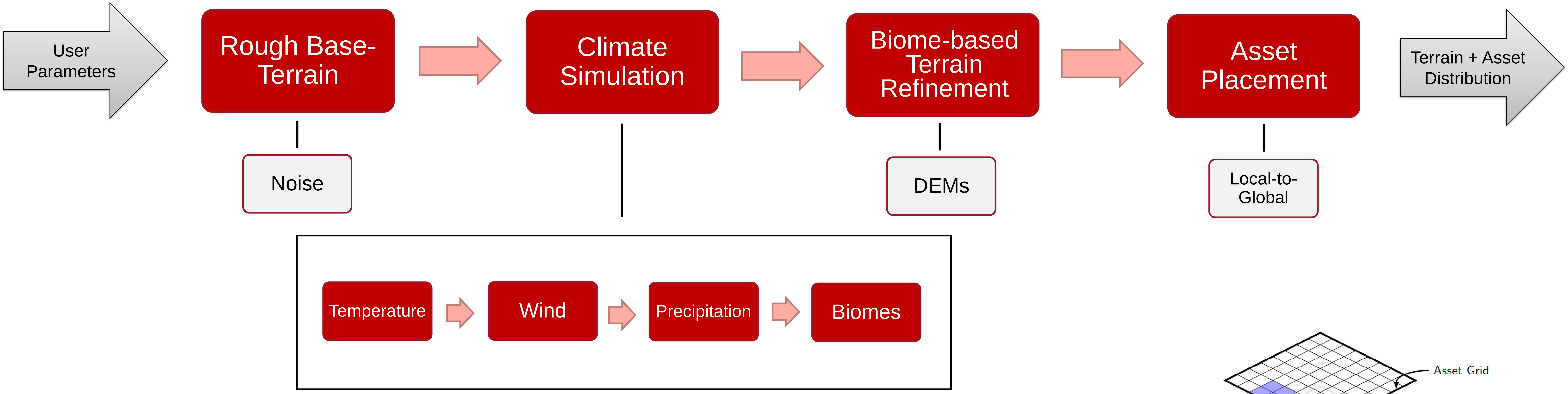
- Each step is customizable and repeatable

Our PTG Pipeline

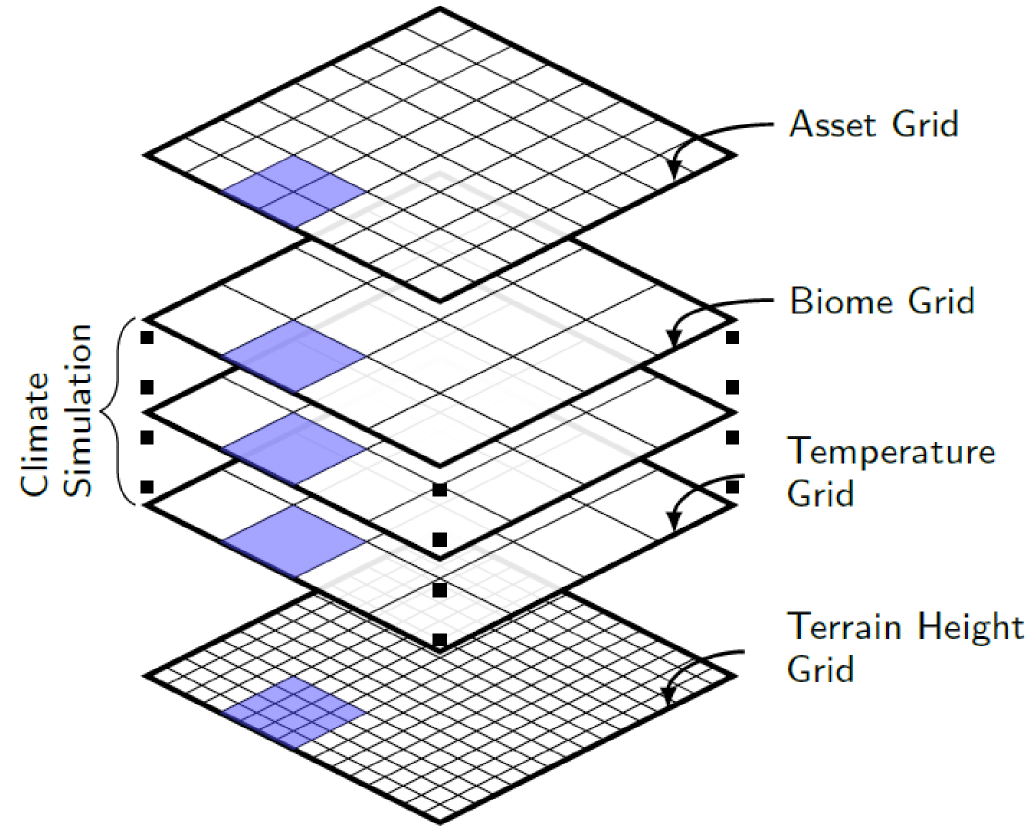


- Each step is customizable and repeatable
- Direct proxy visualization for quick workflow

Our PTG Pipeline

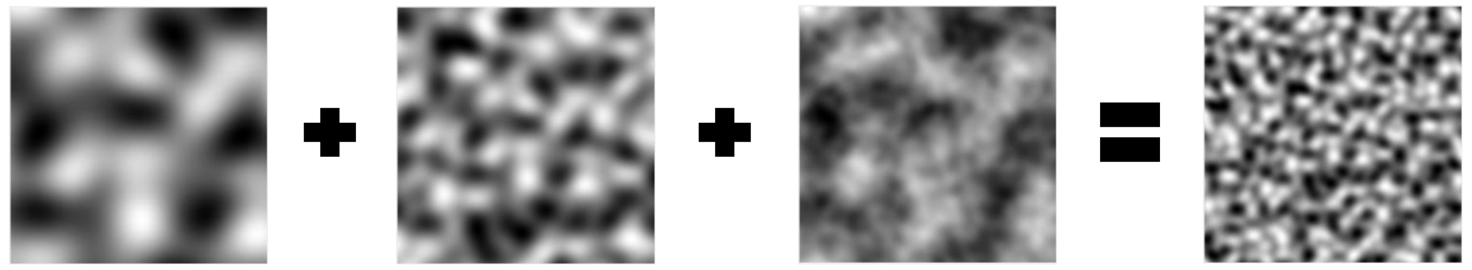


- Each step is customizable and repeatable
- Direct proxy visualization for quick workflow



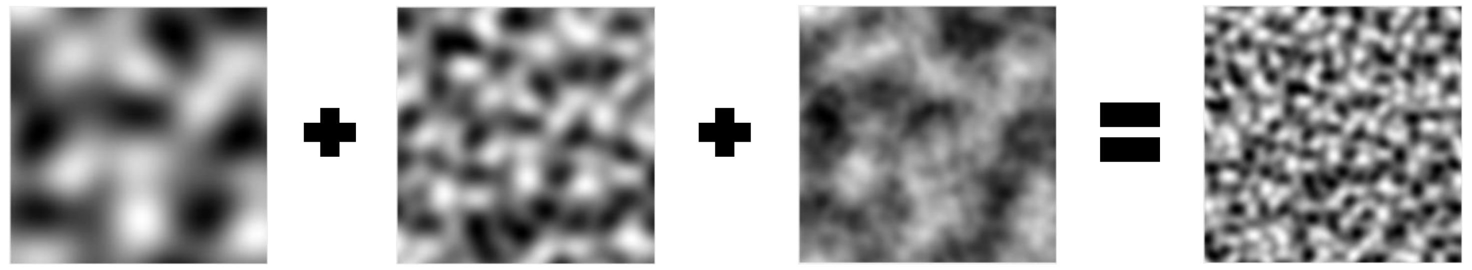
Base Terrain Generation

- Multiple octaves of simplex noise



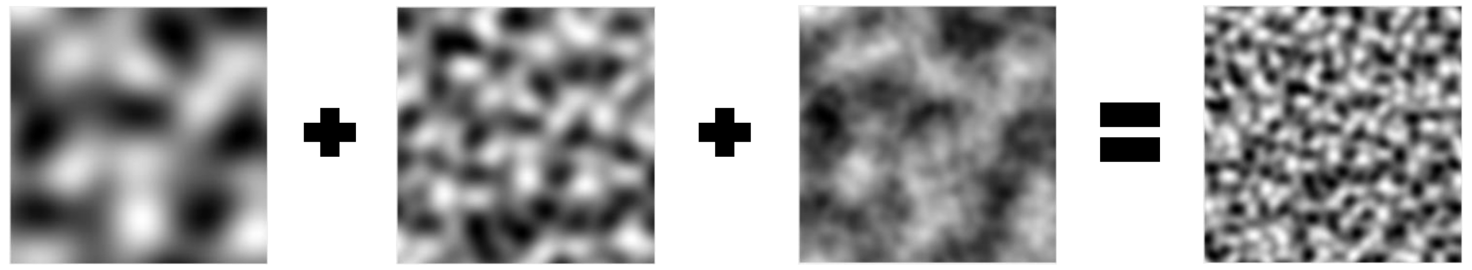
Base Terrain Generation

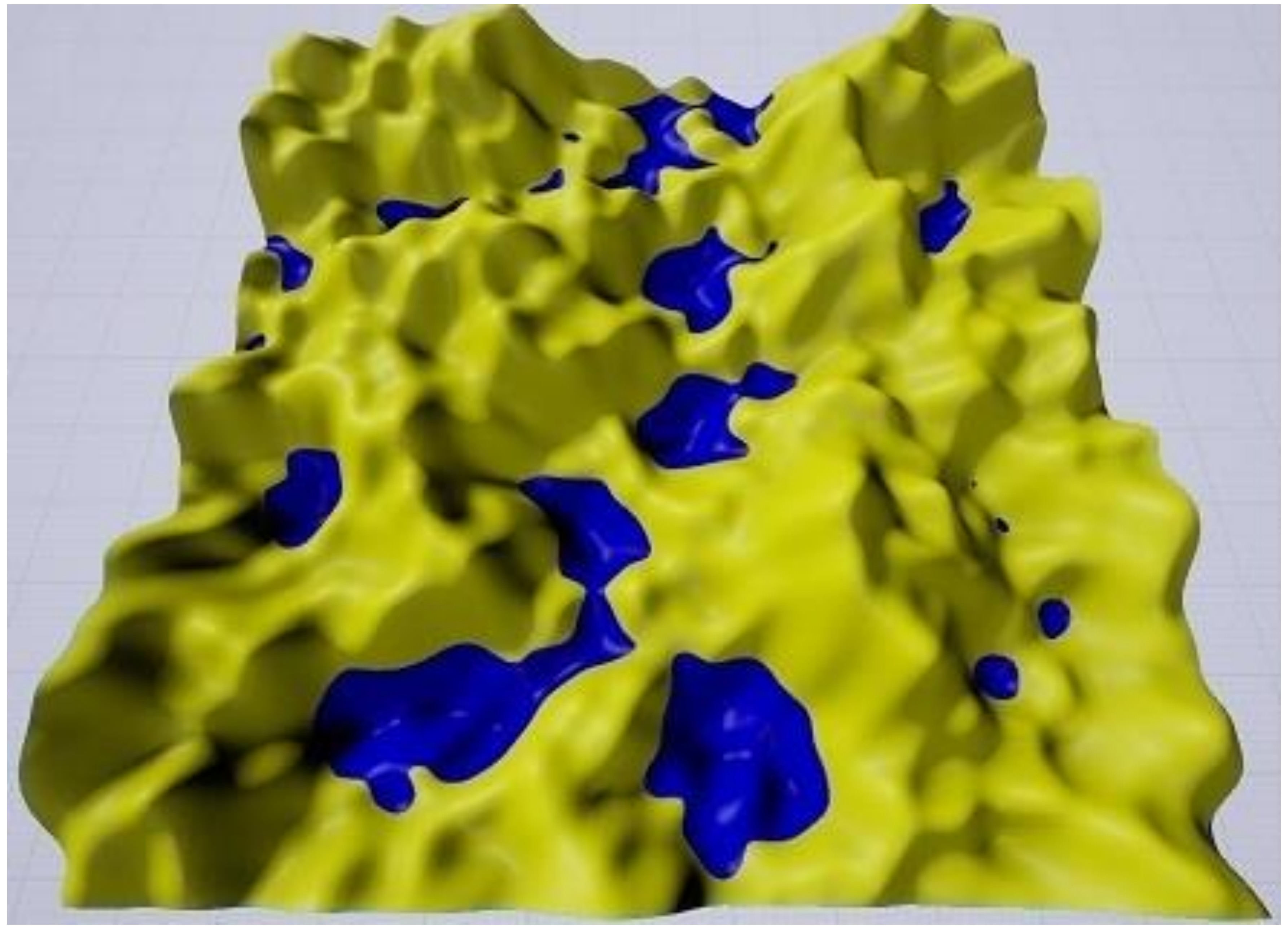
- Multiple octaves of simplex noise
  - Lots of parameters for flexibility, fast to compute, scalable
  - Other noise combinations possible



Base Terrain Generation

- Multiple octaves of simplex noise
  - Lots of parameters for flexibility, fast to compute, scalable
  - Other noise combinations possible
- Only serves as rough starting terrain, refined later
  - No tedious fine-tuning needed

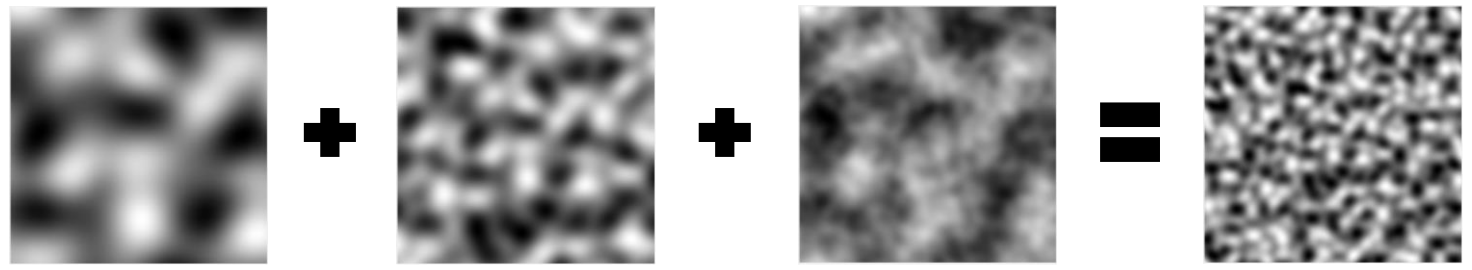


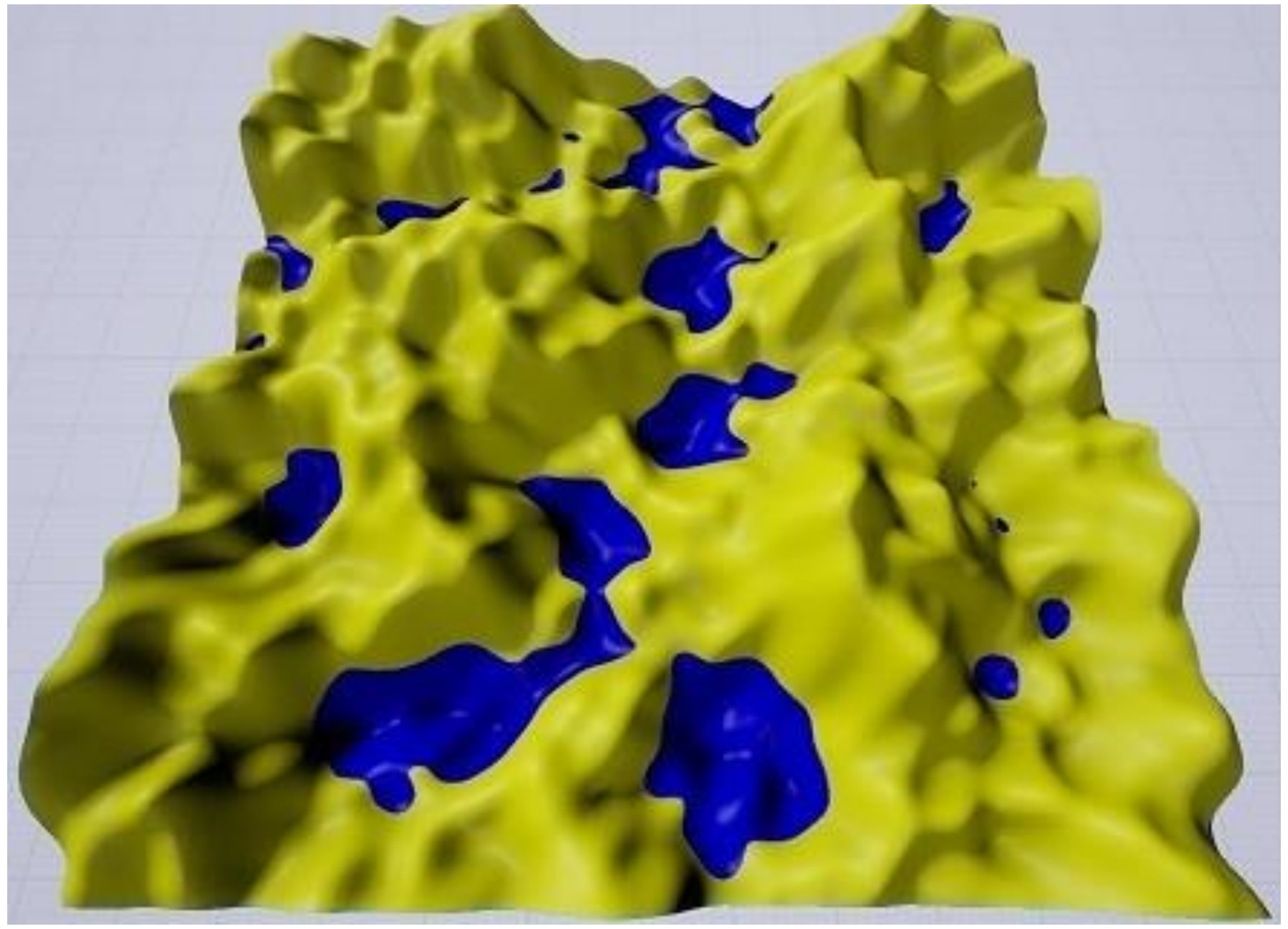


Base terrain, water bodies in blue

Base Terrain Generation

- Multiple octaves of simplex noise
  - Lots of parameters for flexibility, fast to compute, scalable
  - Other noise combinations possible
- Only serves as rough starting terrain, refined later
  - No tedious fine-tuning needed
- Easily extendable with sketch-based editing techniques

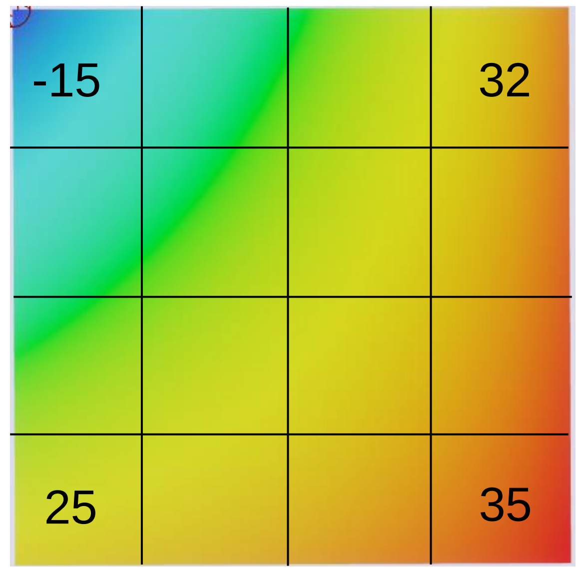




Base terrain, water bodies in blue

Climate Simulation - Temperature

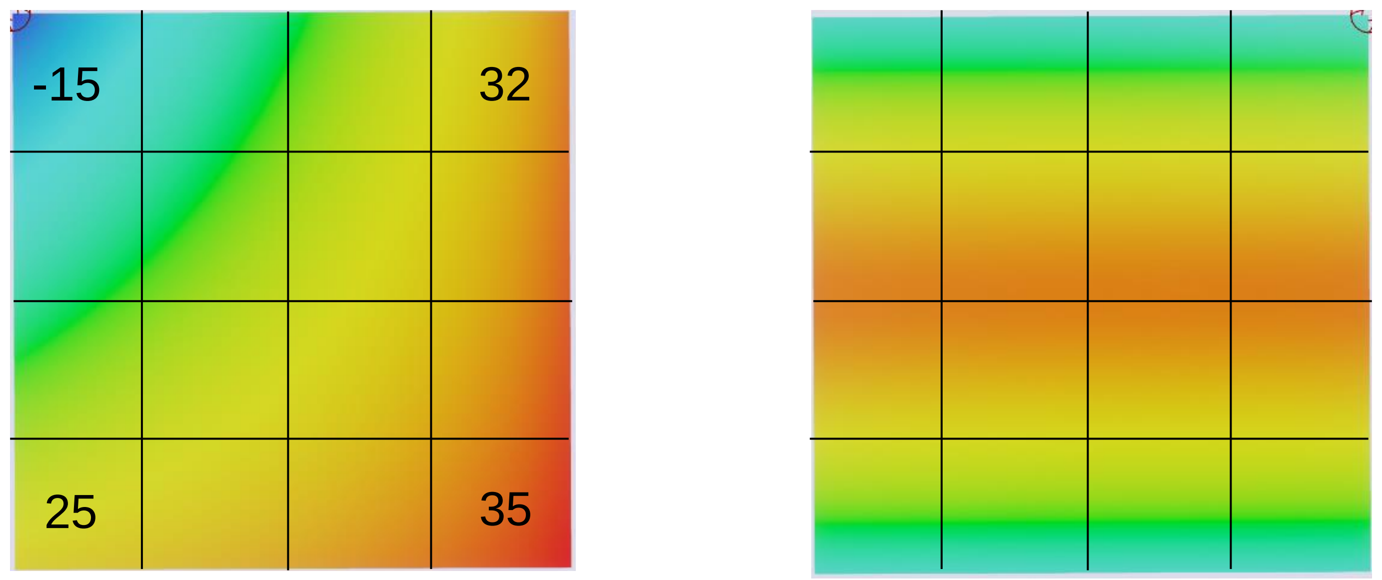
- Two adjustable interpolation modes:
  - Bi-linear interpolation



Temperature, blue = cold, red = hot

Climate Simulation - Temperature

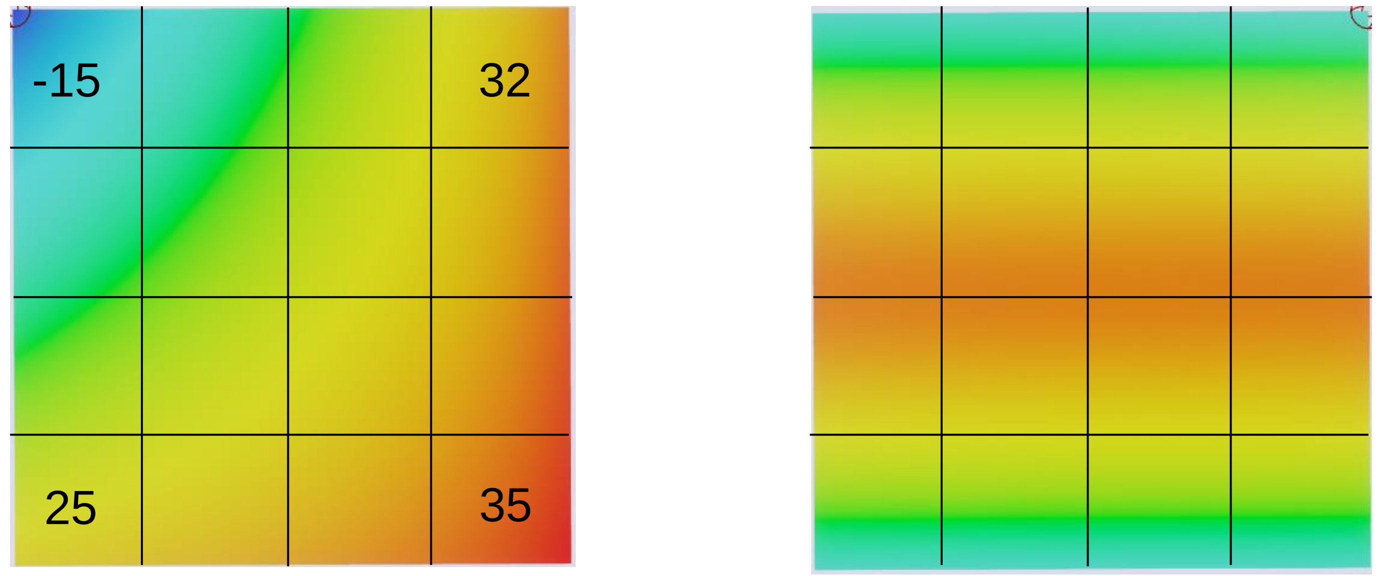
- Two adjustable interpolation modes:
  - Bi-linear interpolation
  - Sine-based interpolation



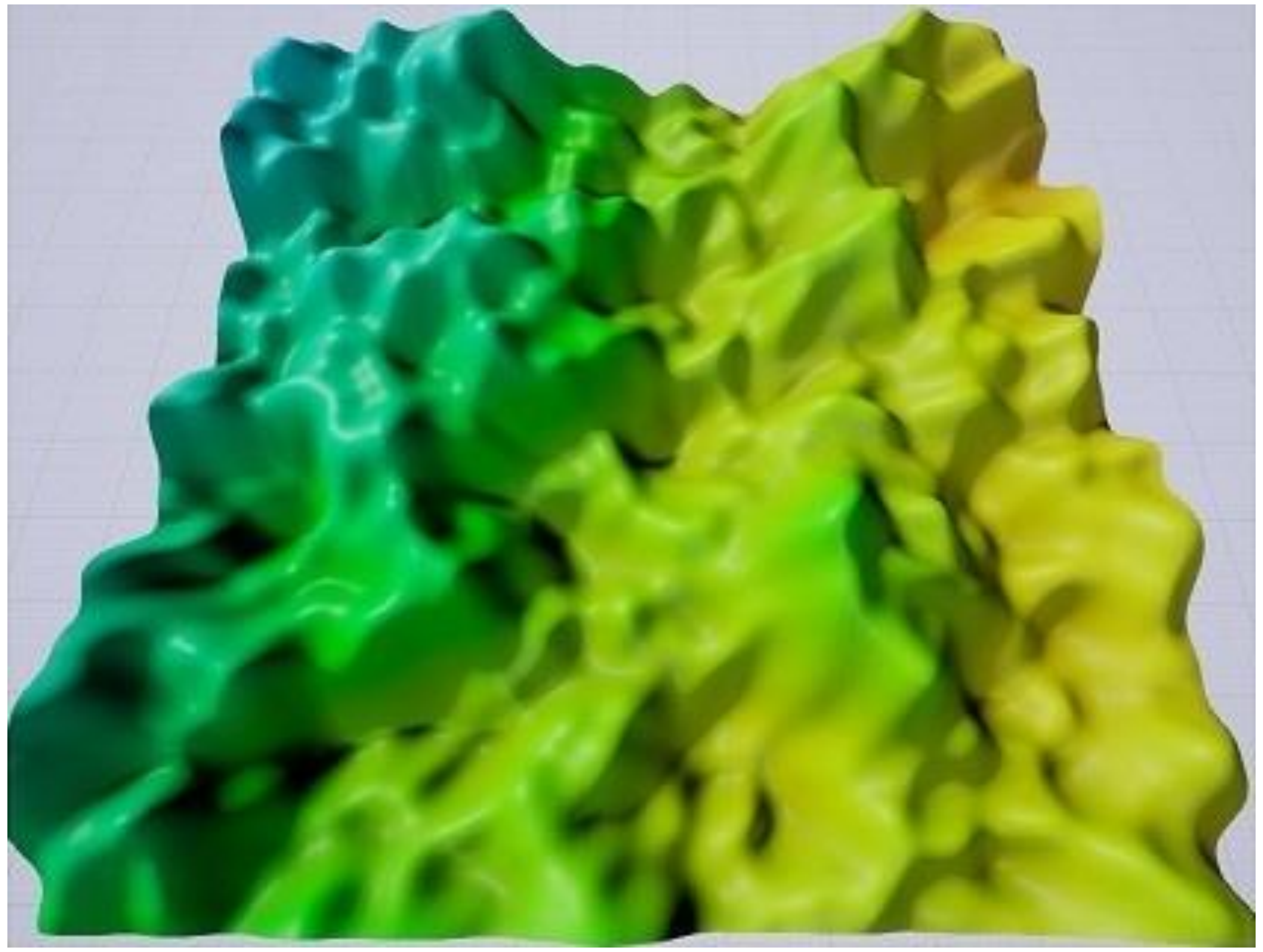
Temperature, blue = cold, red = hot

Climate Simulation - Temperature

- Two adjustable interpolation modes:
  - Bi-linear interpolation
  - Sine-based interpolation
- Adjustable altitude-based decline



Temperature, blue = cold, red = hot



Climate Simulation - Wind

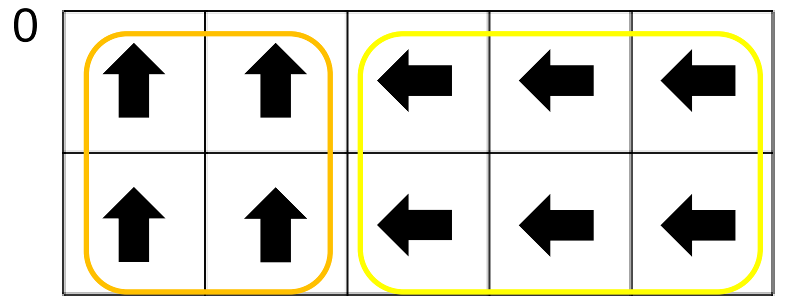
- Prevailing wind for moisture distribution
- Iterative, simplified semi-Lagrangian approach

Climate Simulation - Wind

- Prevailing wind for moisture distribution
- Iterative, simplified semi-Lagrangian approach
  - Only self advection and external forces

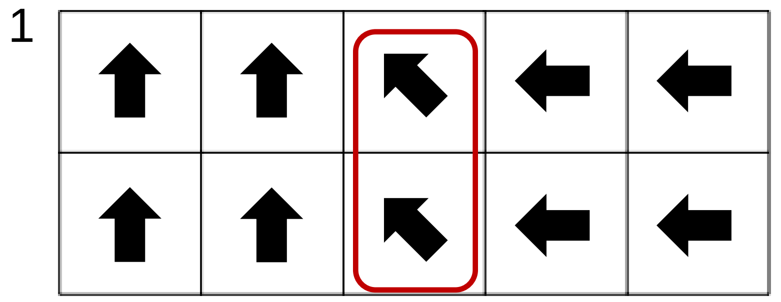
Climate Simulation - Wind

- Prevailing wind for moisture distribution
- Iterative, simplified semi-Lagrangian approach
  - Only self advection and external forces
    - User specified, persistent external forces on corners



Climate Simulation - Wind

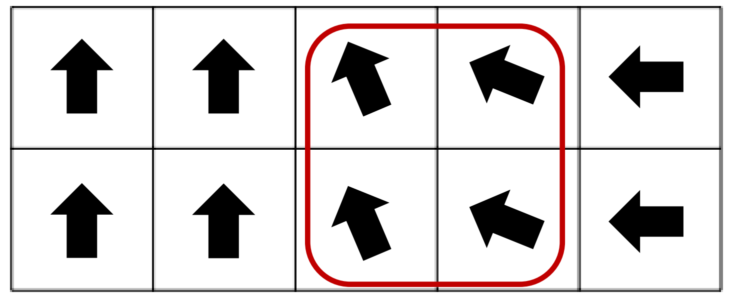
- Prevailing wind for moisture distribution
- Iterative, simplified semi-Lagrangian approach
  - Only self advection and external forces
    - User specified, persistent external forces on corners
    - Iteratively averaging wind vectors with adjacent ones in forward direction



Climate Simulation - Wind

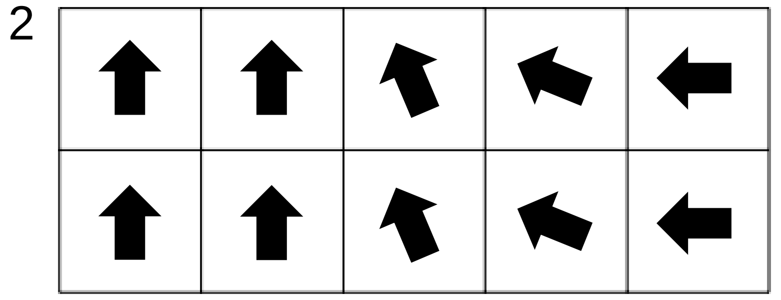
- Prevailing wind for moisture distribution
- Iterative, simplified semi-Lagrangian approach
  - Only self advection and external forces
    - User specified, persistent external forces on corners
    - Iteratively averaging wind vectors with adjacent ones in forward direction

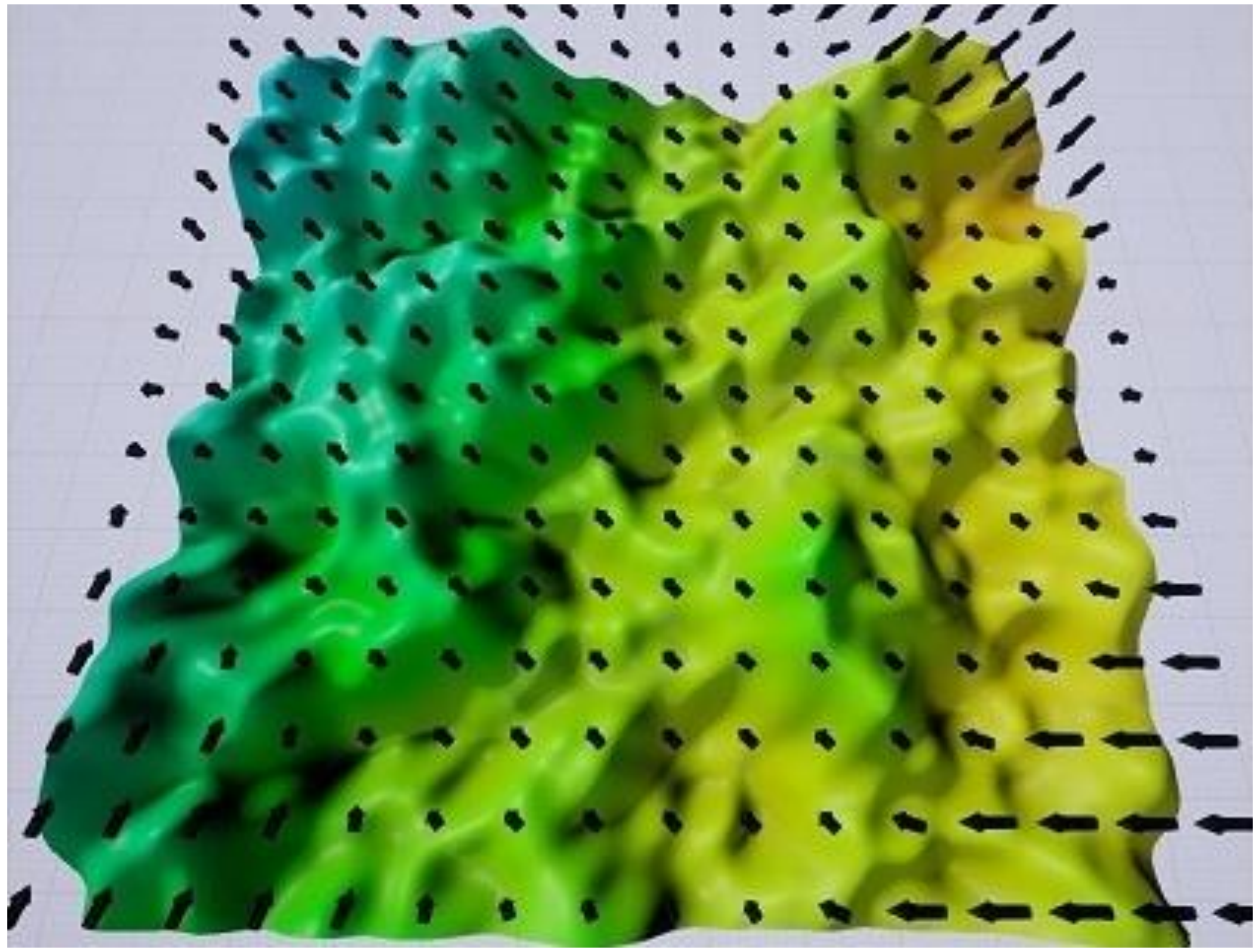
2



Climate Simulation - Wind

- Prevailing wind for moisture distribution
- Iterative, simplified semi-Lagrangian approach
  - Only self advection and external forces
    - User specified, persistent external forces on corners
    - Iteratively averaging wind vectors with adjacent ones in forward direction
  - Enables creation of smooth, believable prevailing wind currents





Wind vector field

Climate Simulation - Precipitation

- Iterative computation

Climate Simulation - Precipitation

- Iterative computation
- Water bodies/world borders as moisture sources

Climate Simulation - Precipitation

- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation

Climate Simulation - Precipitation

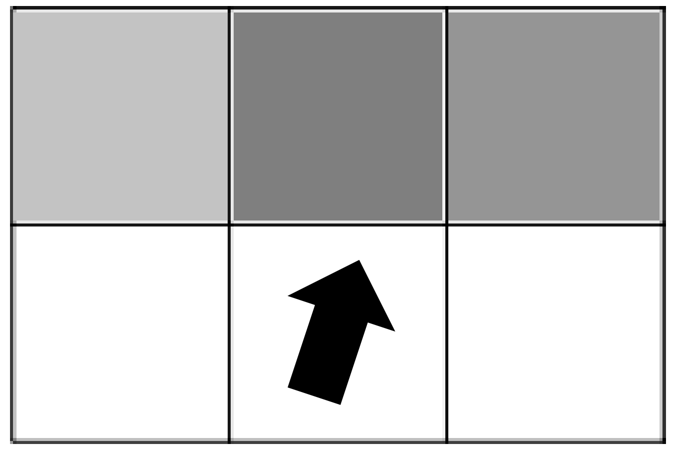
- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture

Climate Simulation - Precipitation

- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture
  - Dispersion and equalization

Climate Simulation - Precipitation

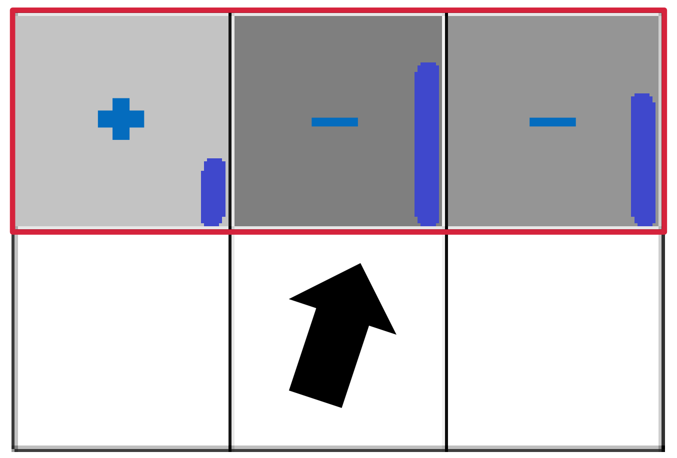
- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture
  - Dispersion and equalization



Grey indicates receiving moisture amount

Climate Simulation - Precipitation

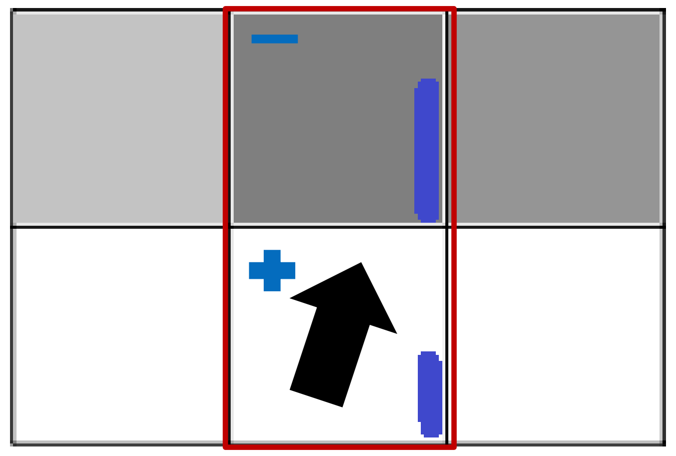
- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture
  - Dispersion and equalization



Diffusion based on current moisture

Climate Simulation - Precipitation

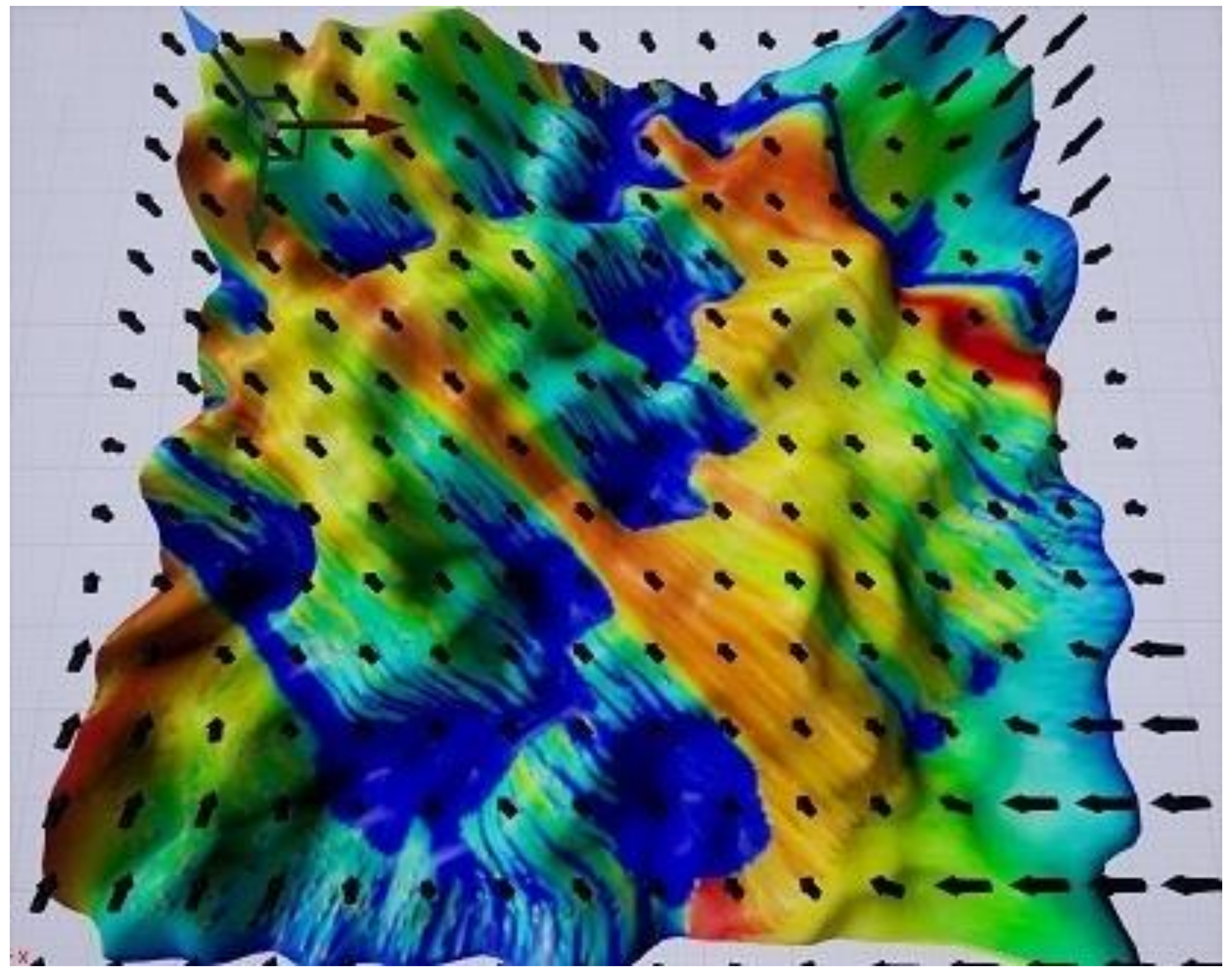
- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture
  - Dispersion and equalization



Diffusion based on current moisture

Climate Simulation - Precipitation

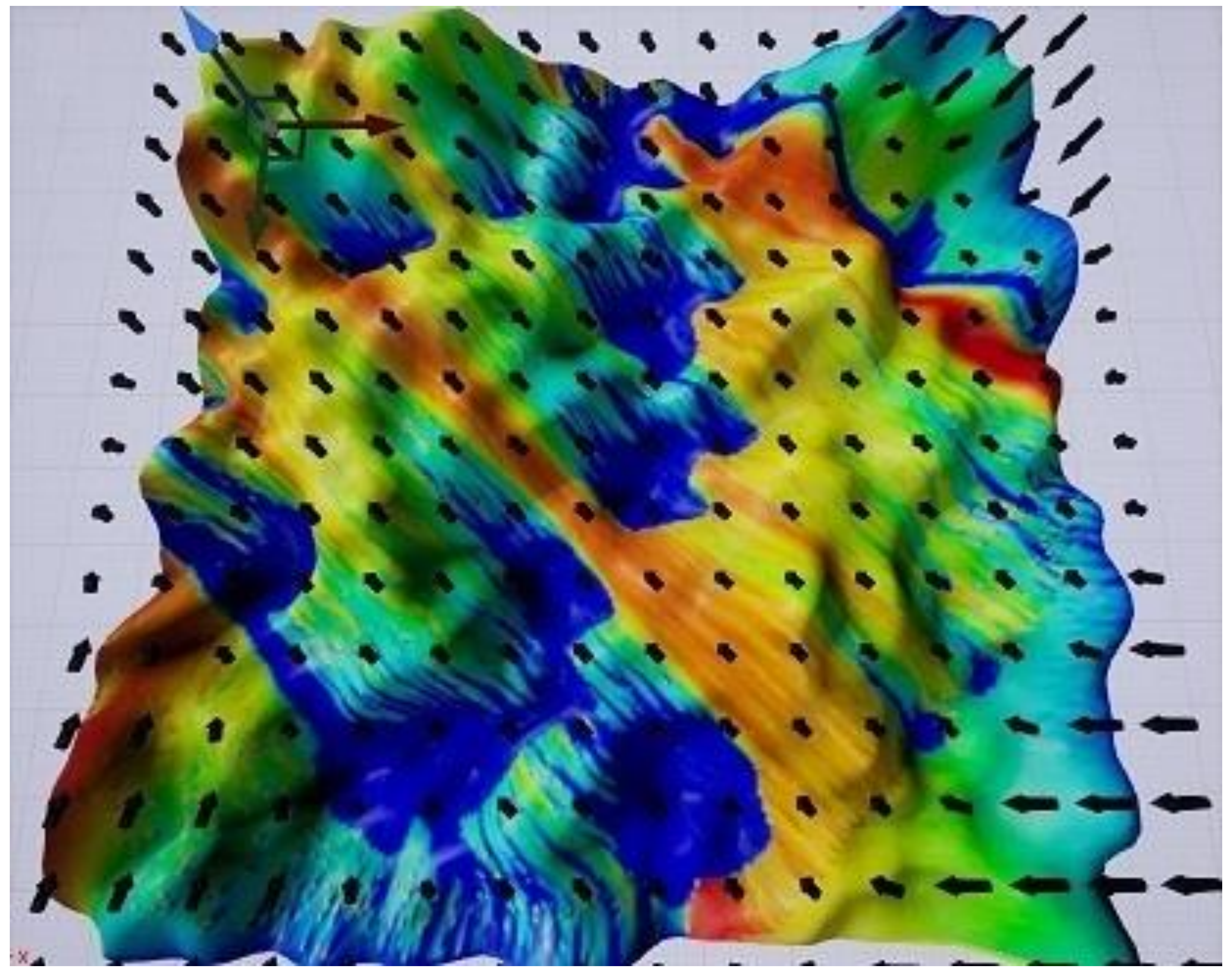
- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture
  - Dispersion and equalization



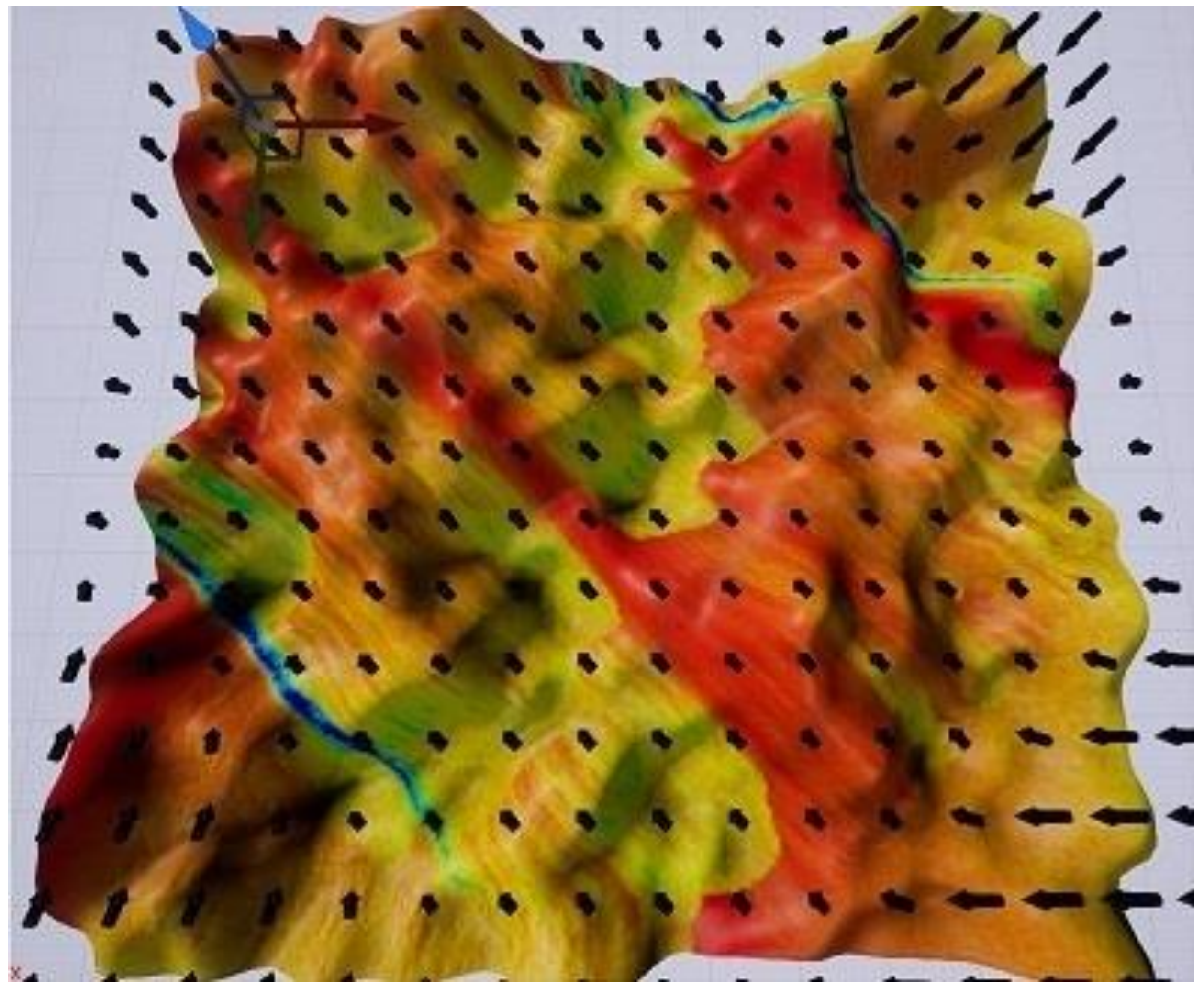
Moisture, low = red, high = blue

Climate Simulation - Precipitation

- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture
  - Dispersion and equalization
- Temperature and moisture-dependent precipitation



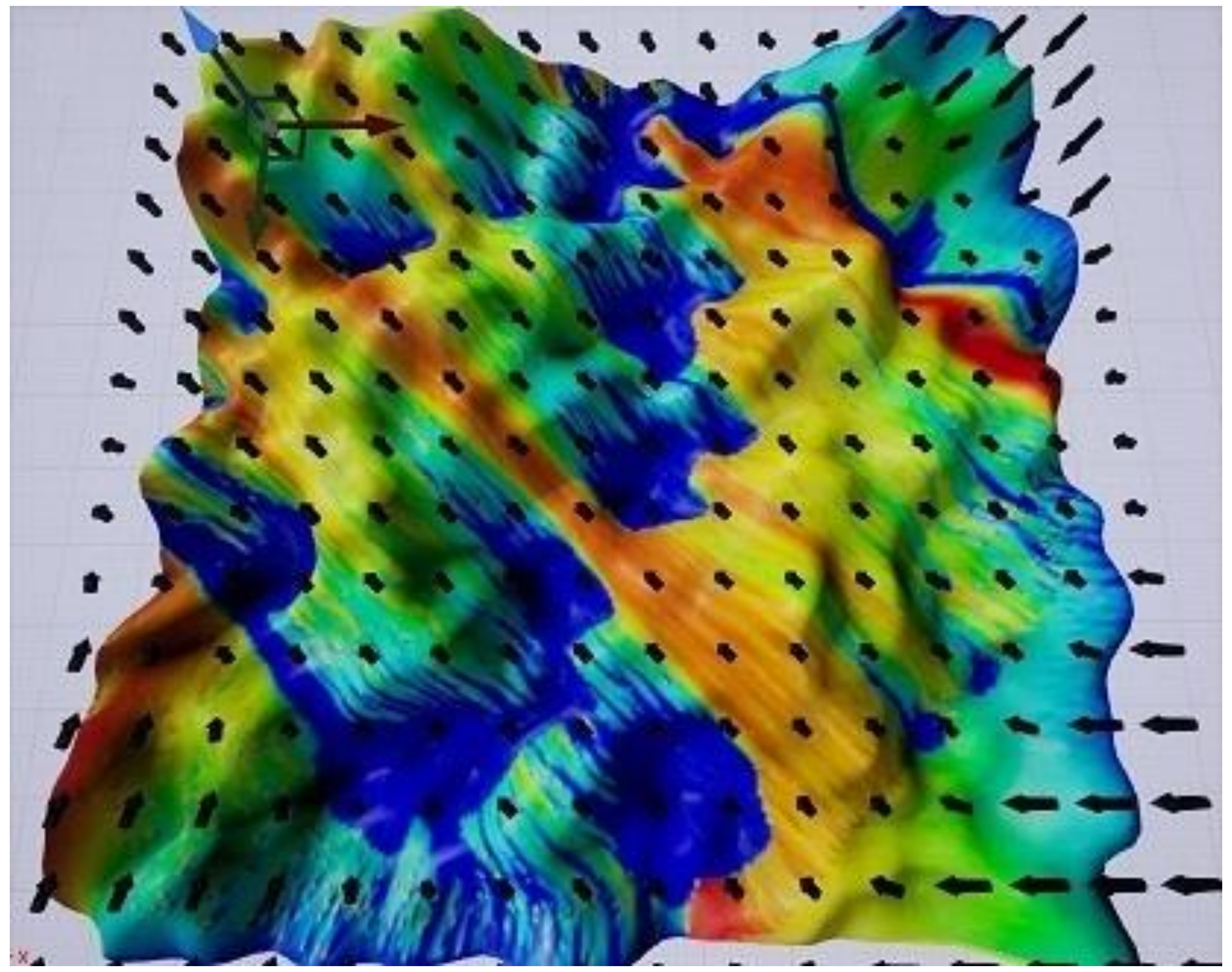
Moisture, low = red, high = blue



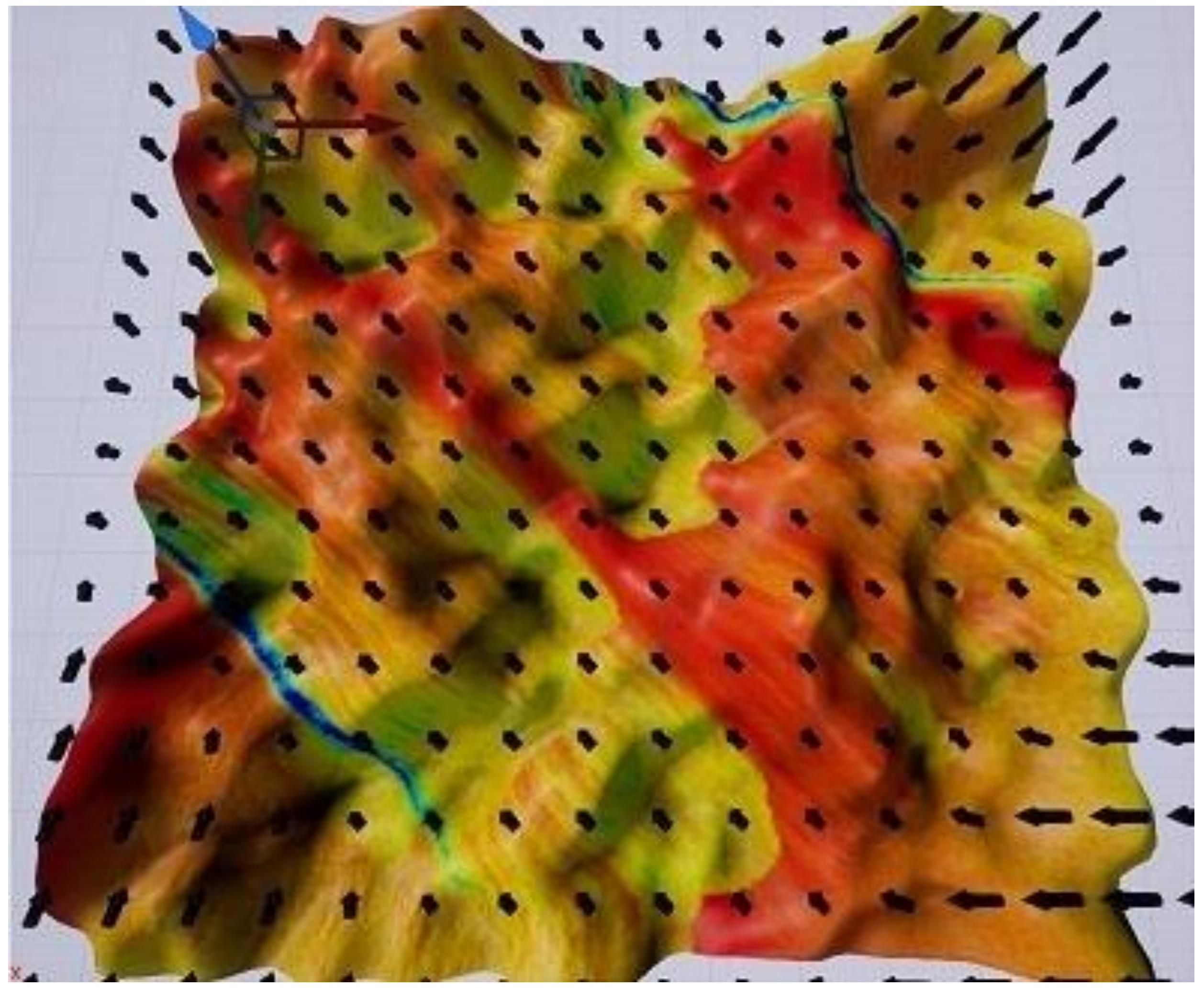
Precipitation, low = red, high = blue

Climate Simulation - Precipitation

- Iterative computation
- Water bodies/world borders as moisture sources
- Temperature-dependent evaporation
- Wind distributes moisture
  - Dispersion and equalization
- Temperature and moisture-dependent precipitation
- Enables phenomena like rain shadows



Moisture, low = red, high = blue



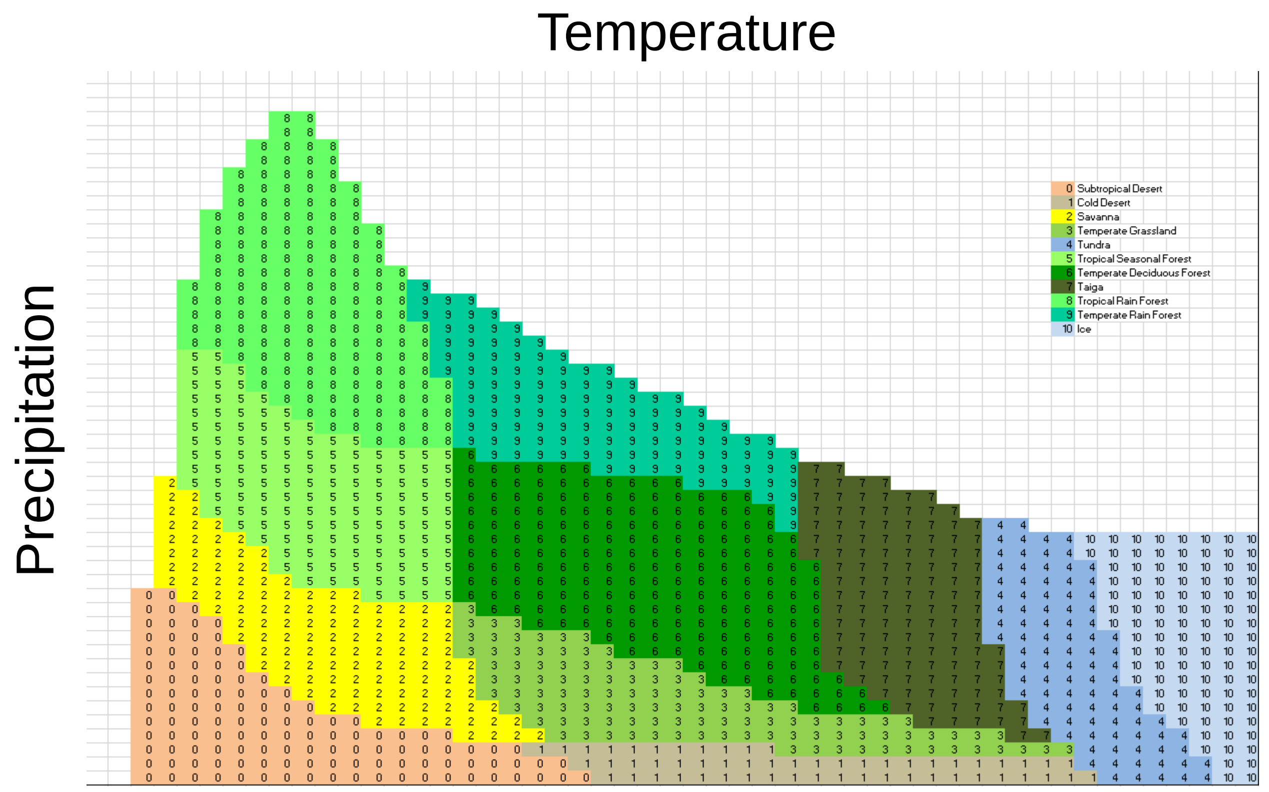
Precipitation, low = red, high = blue

Climate Simulation - Biomes

- Biomes classified by temperature and precipitation

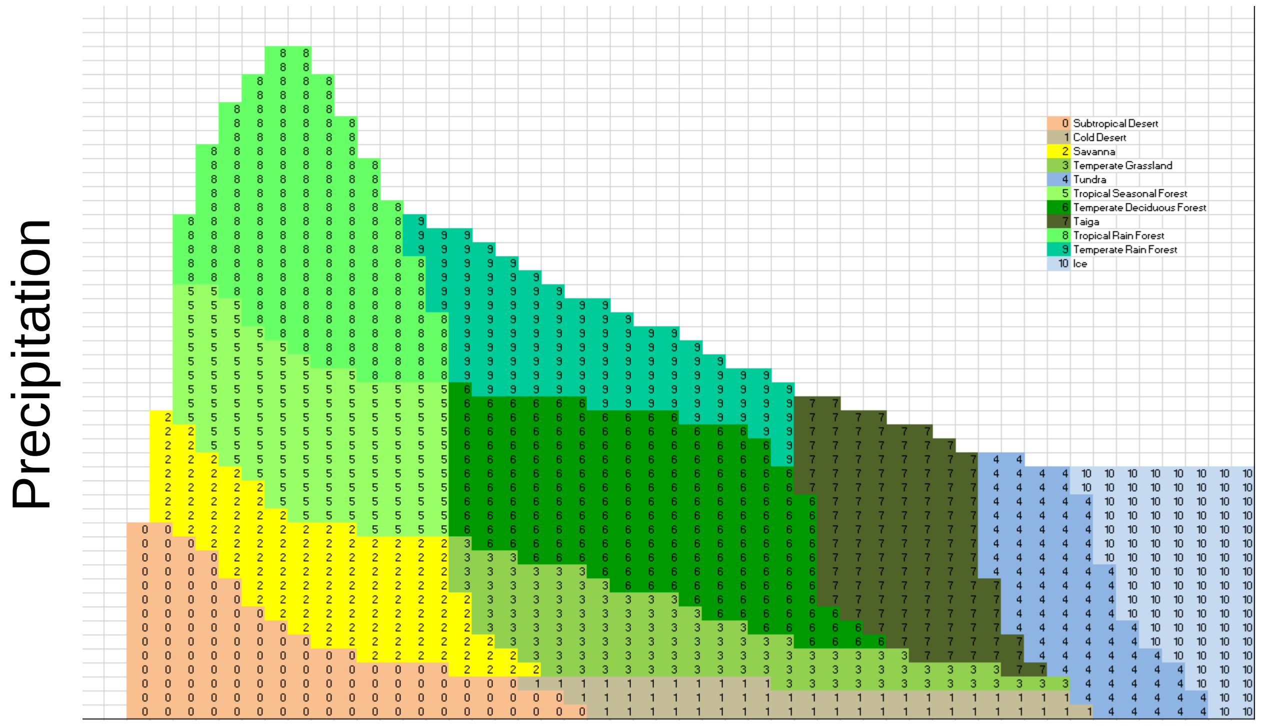
Climate Simulation - Biomes

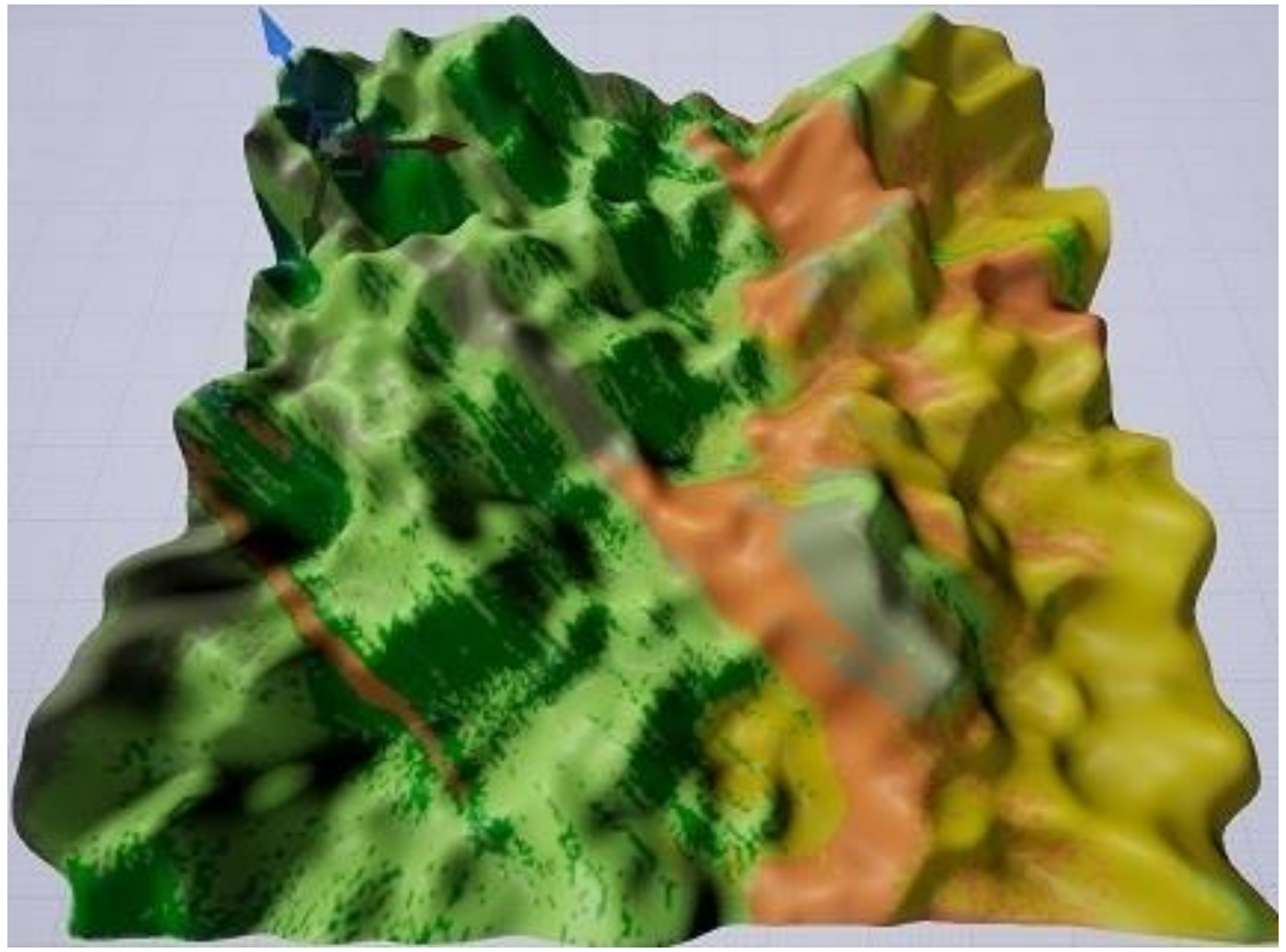
- Biomes classified by temperature and precipitation
- Discretized Whittaker diagram as lookup table
  - Fully customizable or replaceable



Climate Simulation - Biomes

- Biomes classified by temperature and precipitation
- Discretized Whittaker diagram as lookup table
  - Fully customizable or replaceable

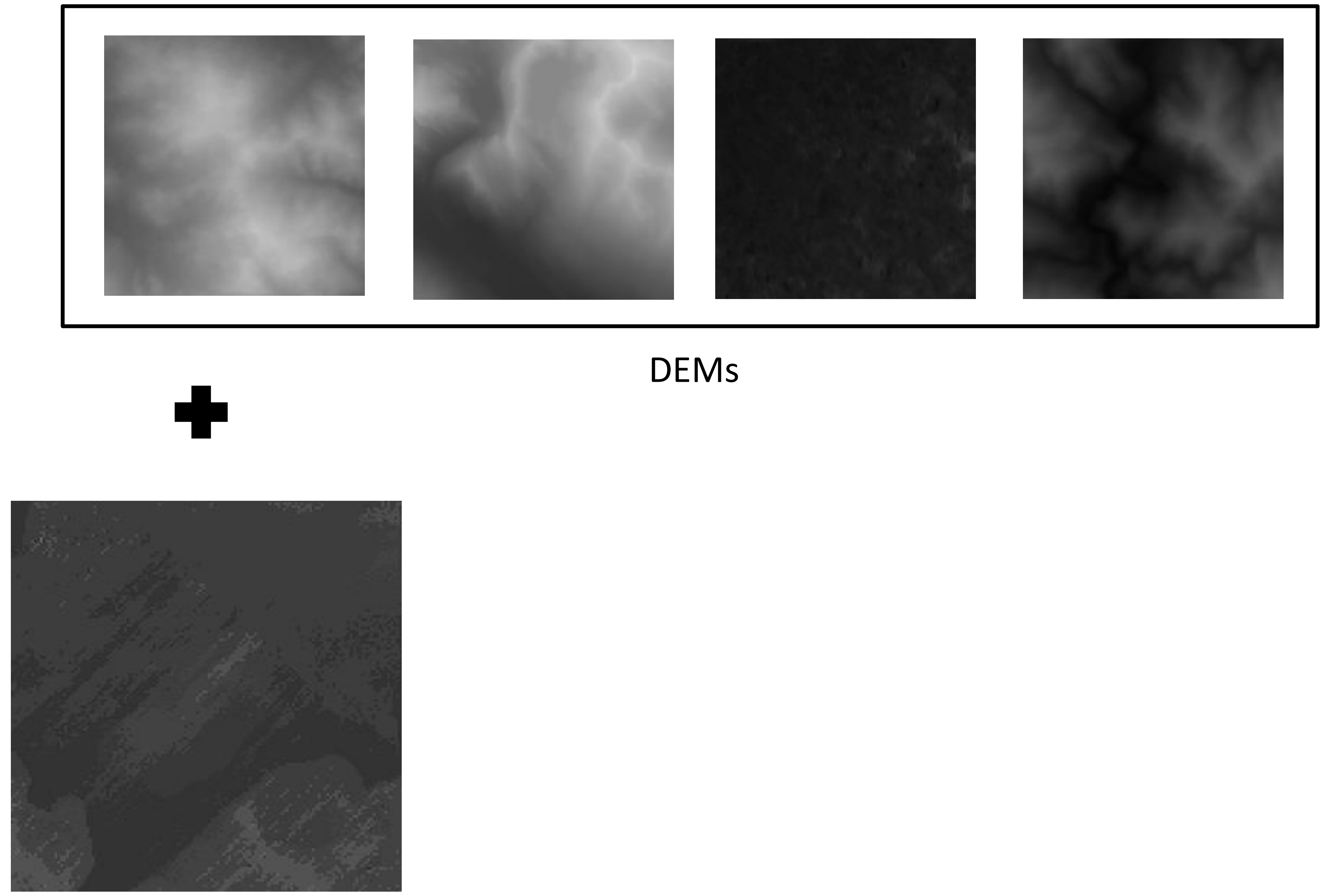




Biome map

Biome-Based Terrain Refinement

- Adds biome-specific structures
- DEMs as examples, inherently realistic

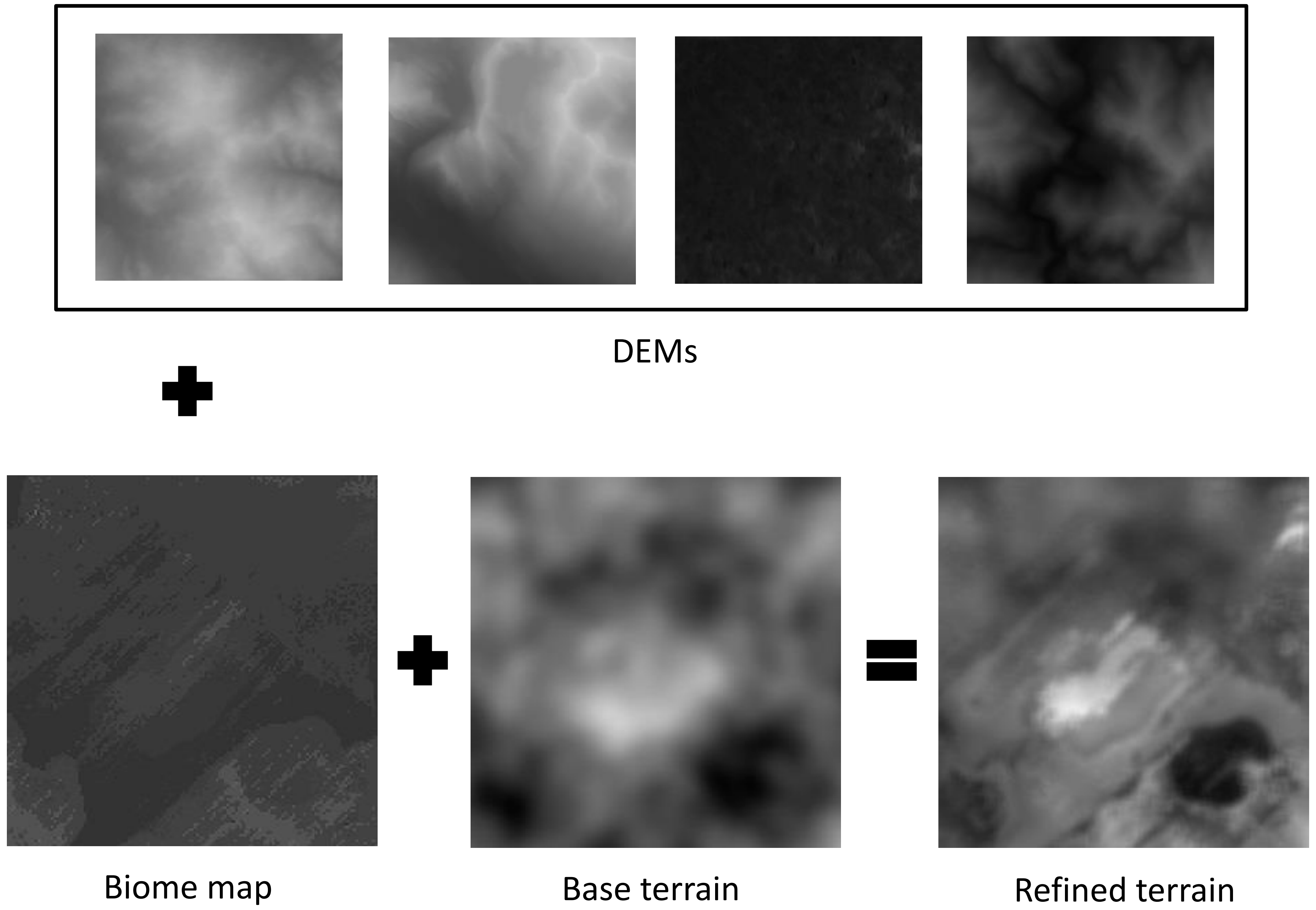


DEMs

Biome map

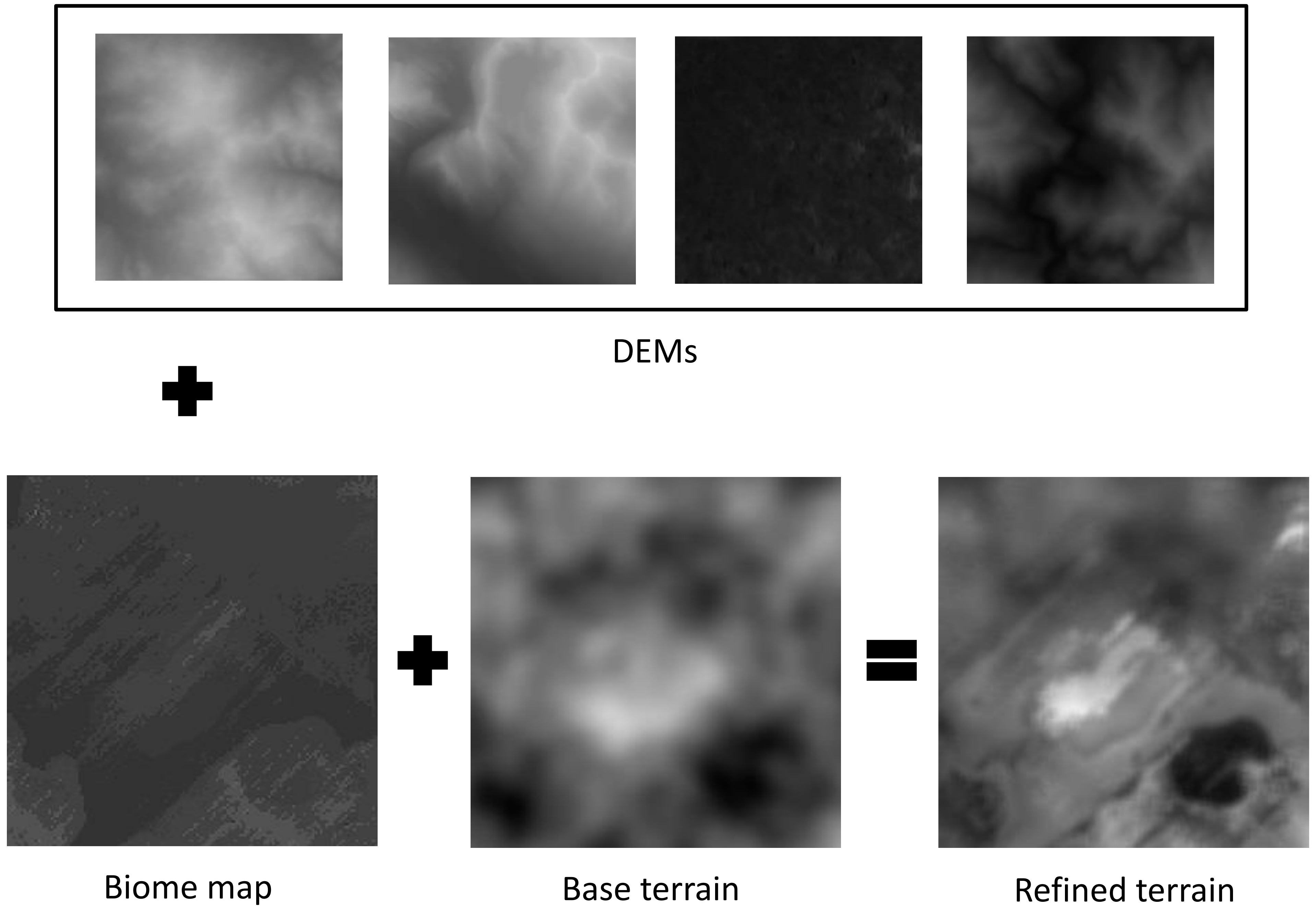
Biome-Based Terrain Refinement

- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b



Biome-Based Terrain Refinement

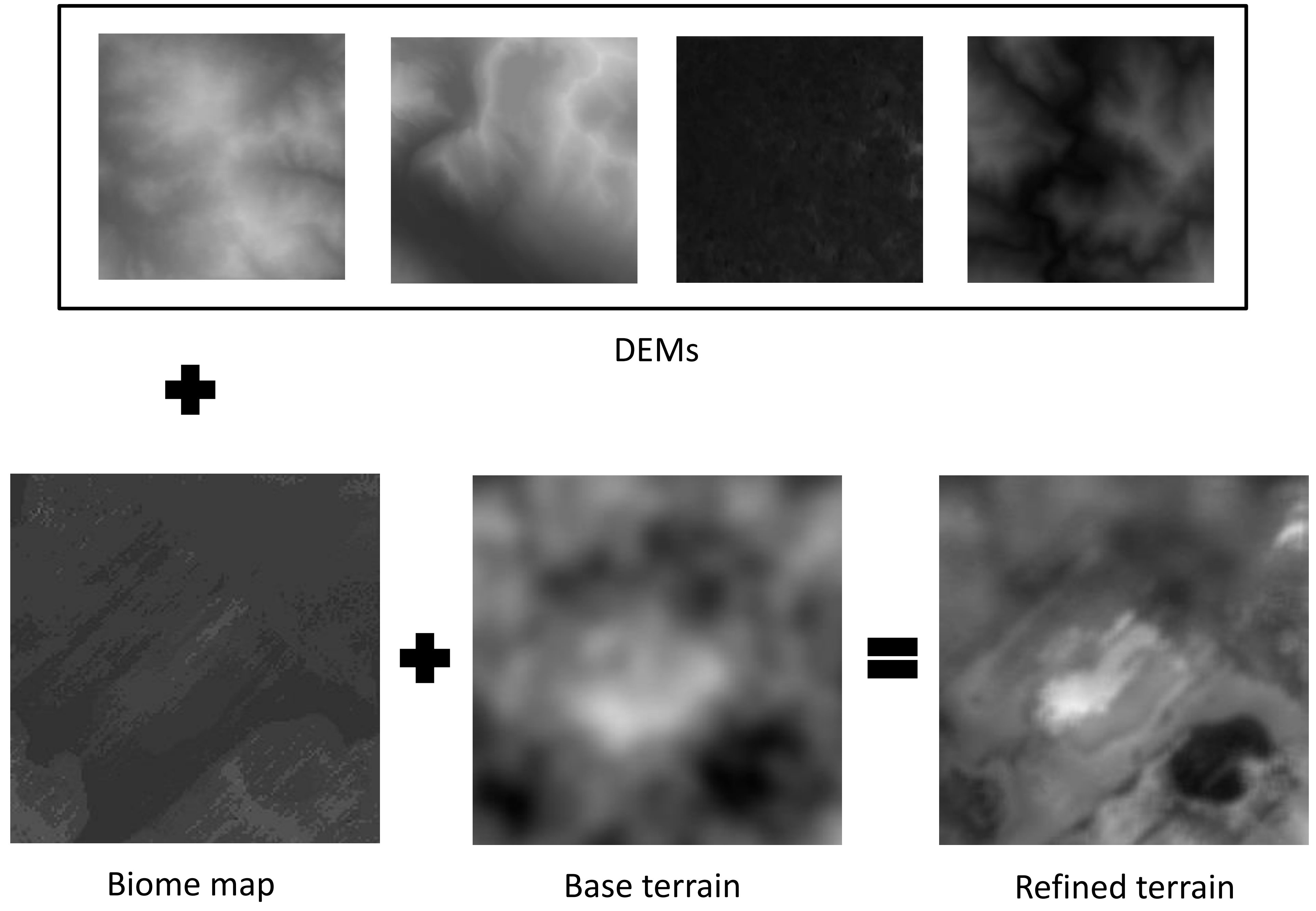
- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b



$$h(p) = w_b \cdot h_b(p) + w_d \cdot h_d(p)$$

Biome-Based Terrain Refinement

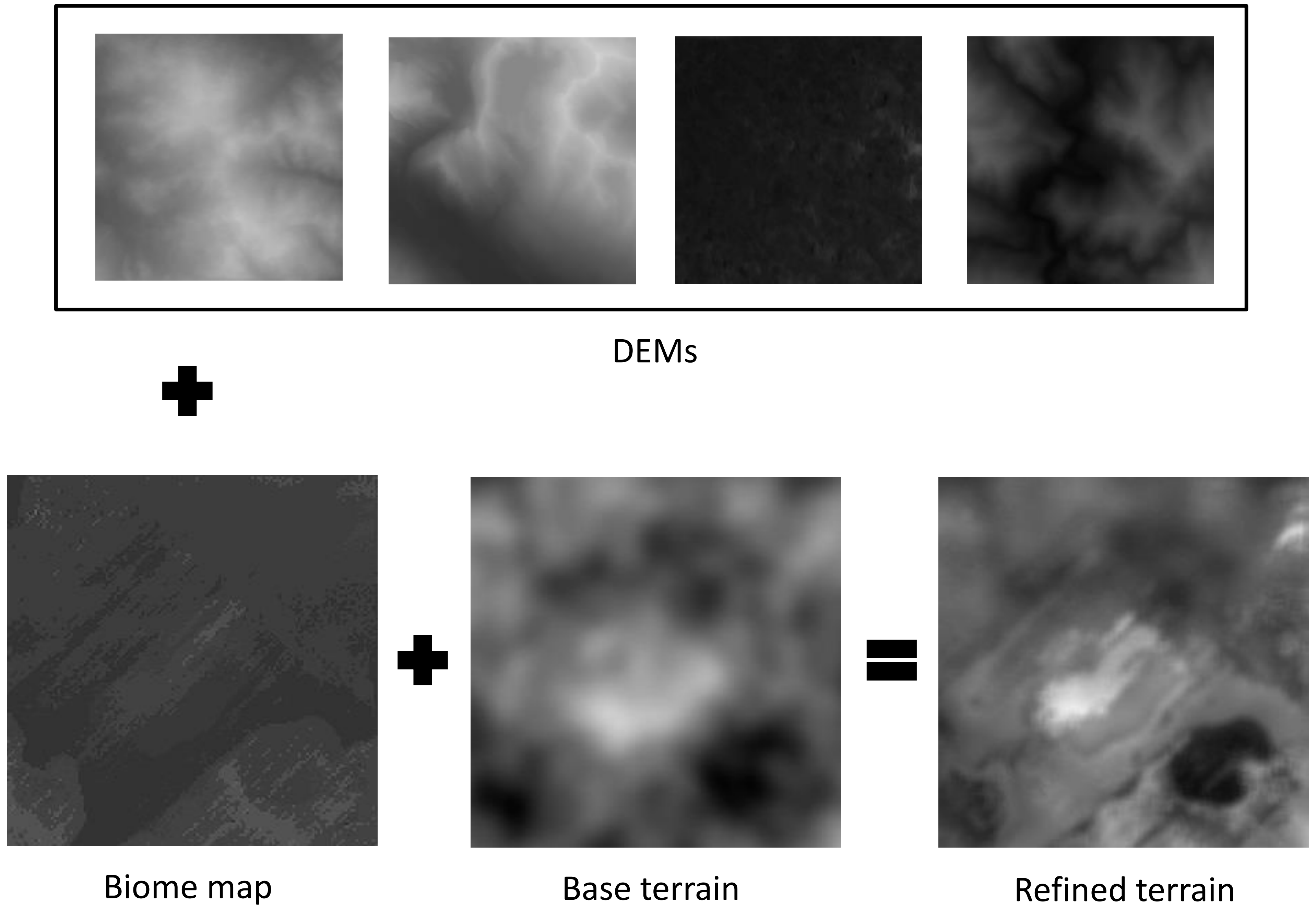
- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b



$$h(p) = \boxed{w_b \cdot h_b(p)} + w_d \cdot h_d(p)$$

Biome-Based Terrain Refinement

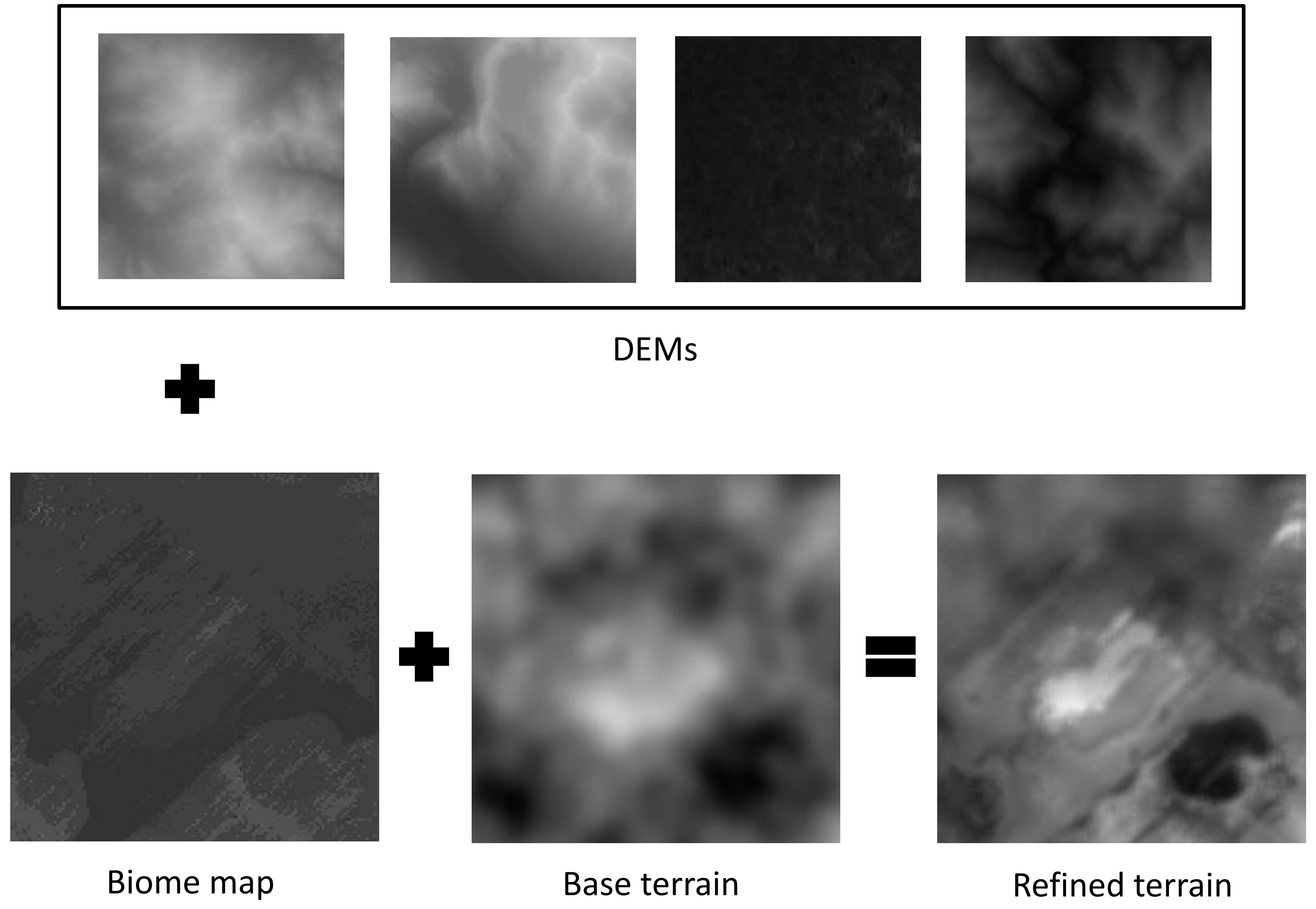
- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b



$$h(p) = w_b \cdot h_b(p) + \boxed{w_d \cdot h_d(p)}$$

Biome-Based Terrain Refinement

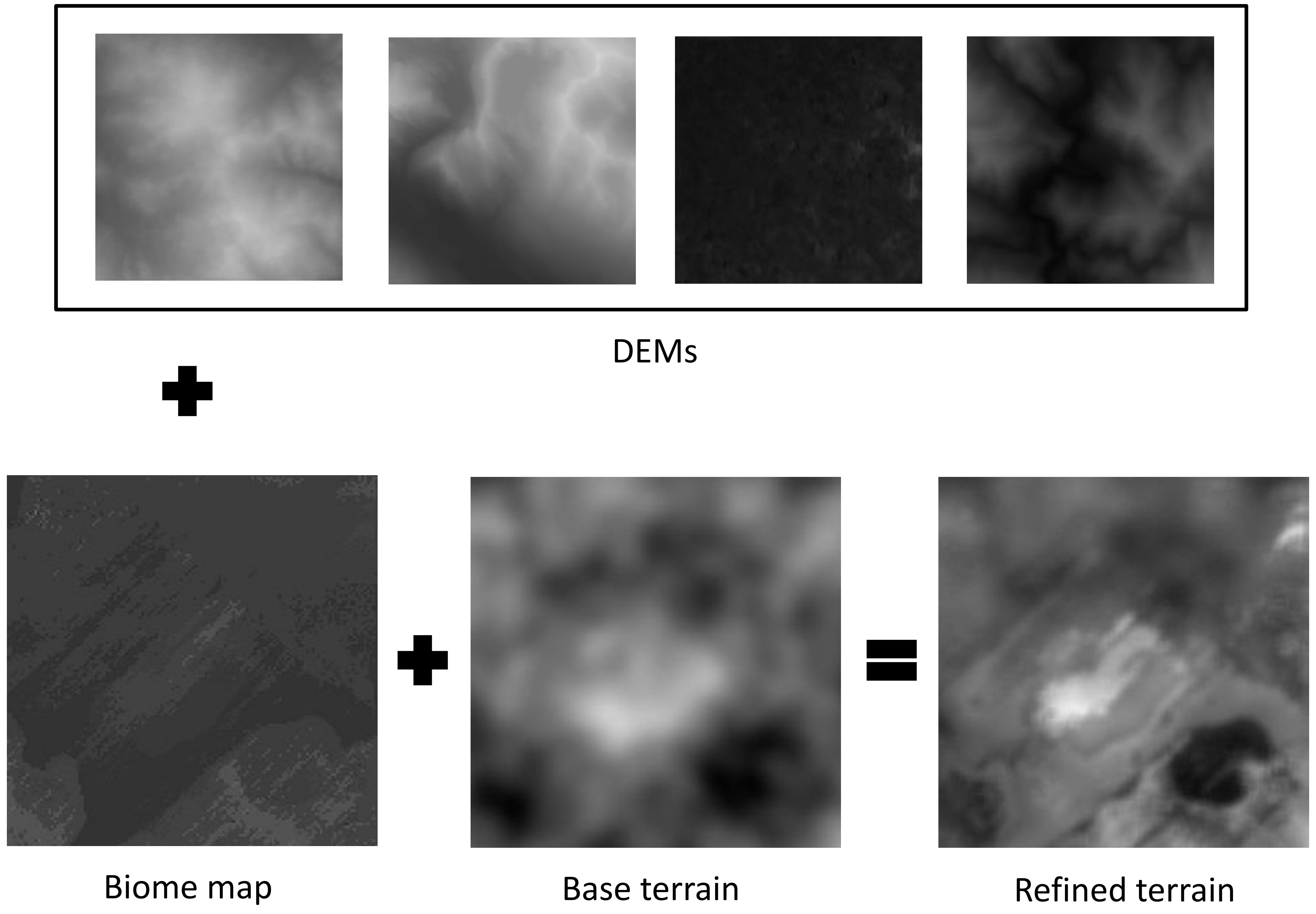
- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b
- Natural biome transitions essential



$$h(p) = w_b \cdot h_b(p) + w_d \cdot h_d(p)$$

Biome-Based Terrain Refinement

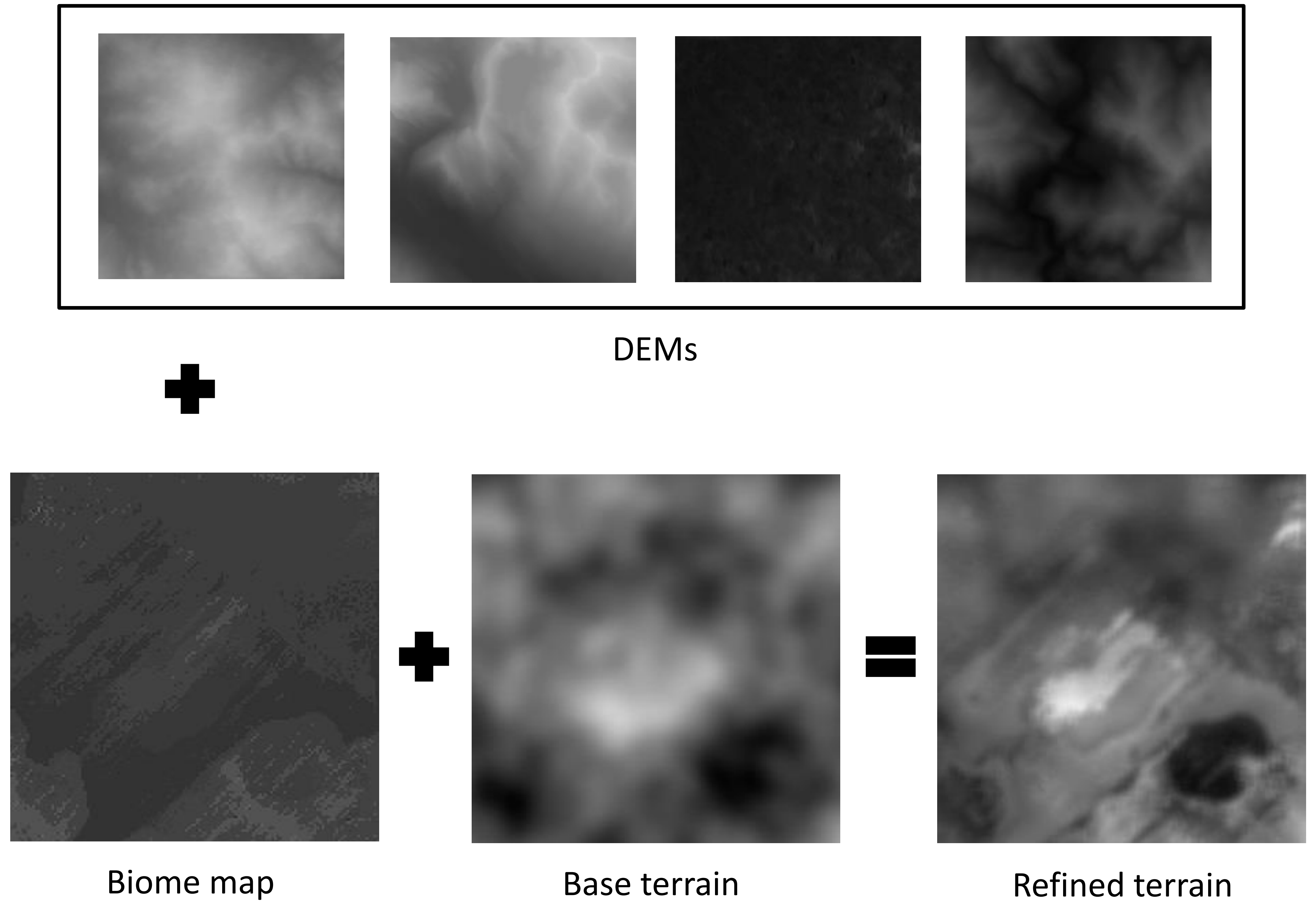
- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b
- Natural biome transitions essential
  - Further noise-based border distortion



$$h(p) = w_b \cdot h_b(p) + w_d \cdot h_d(p)$$

Biome-Based Terrain Refinement

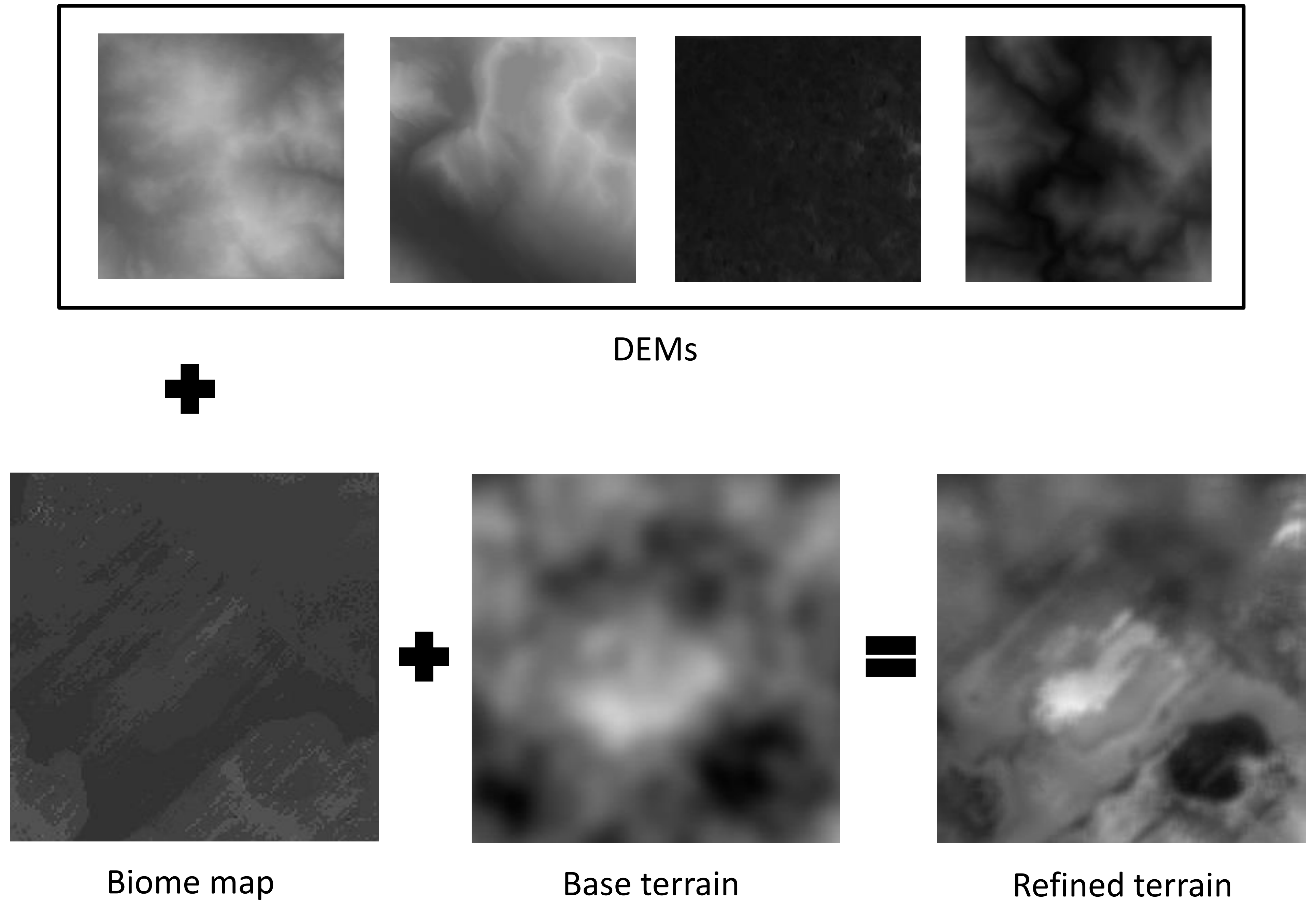
- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b
- Natural biome transitions essential
  - Further noise-based border distortion
  - Weighted blending of adjacent DEMs, i, via 2D kernel



$$h(p) = w_b \cdot h_b(p) + w_d \cdot h_d(p)$$

Biome-Based Terrain Refinement

- Adds biome-specific structures
- DEMs as examples, inherently realistic
- DEMs, h_d, blended with base terrain, h_b
- Natural biome transitions essential
  - Further noise-based border distortion
  - Weighted blending of adjacent DEMs, i, via 2D kernel



$$h(p) = w_b \cdot h_b(p) + w_d \cdot \sum_{i=1}^{n} w_i \cdot h_{di}(p)$$

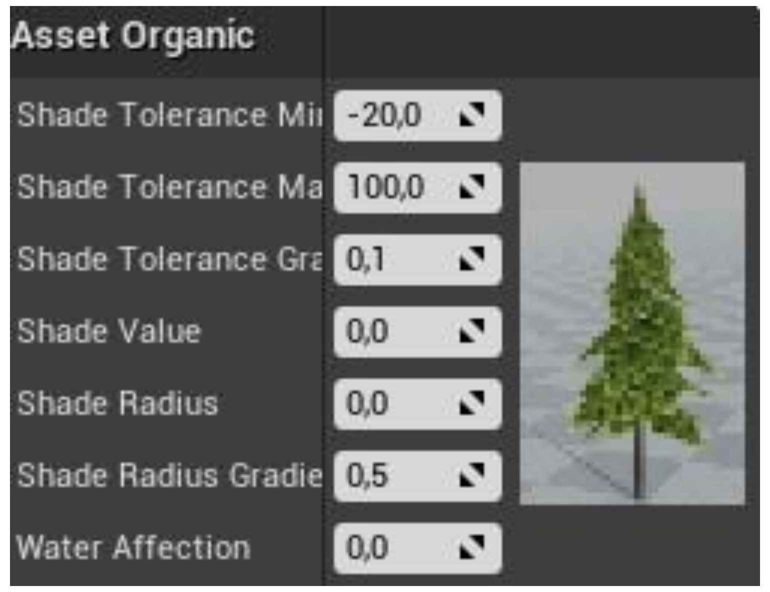
Procedural Asset Placement

- Iterative, rule-based local-to-global model

Bremen

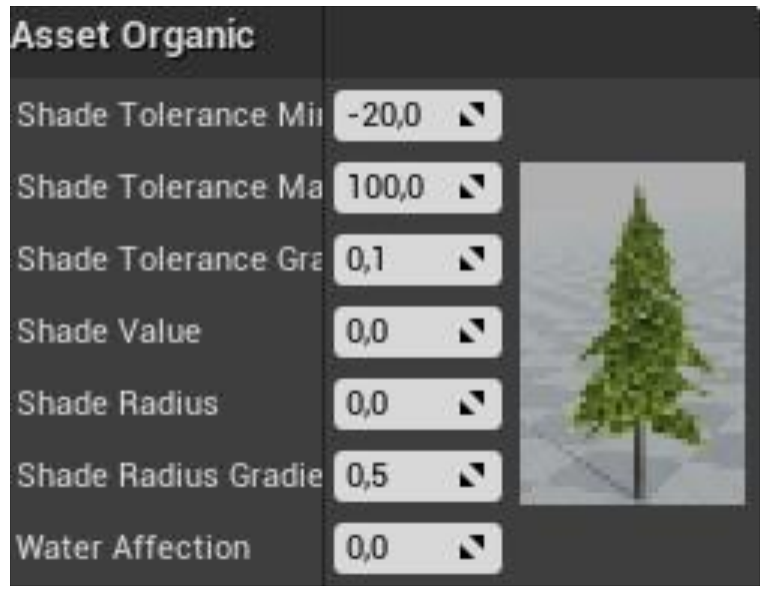
Procedural Asset Placement

- Iterative, rule-based local-to-global model
  - Assets with bilateral placement rules



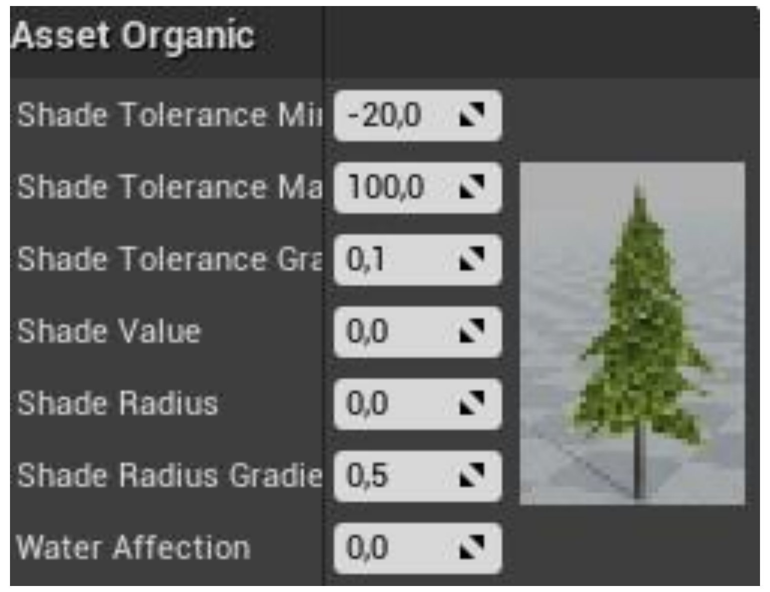
Procedural Asset Placement

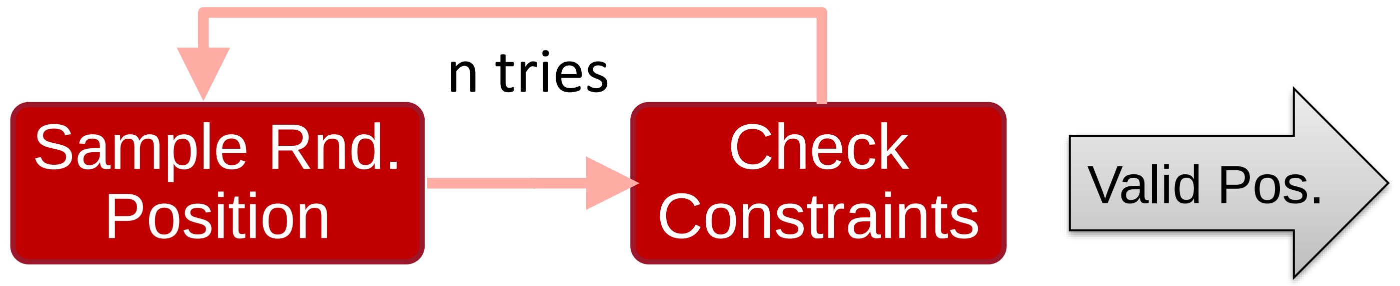
- Iterative, rule-based local-to-global model
  - Assets with bilateral placement rules
  - Asset hierarchy, assigned to biomes



Procedural Asset Placement

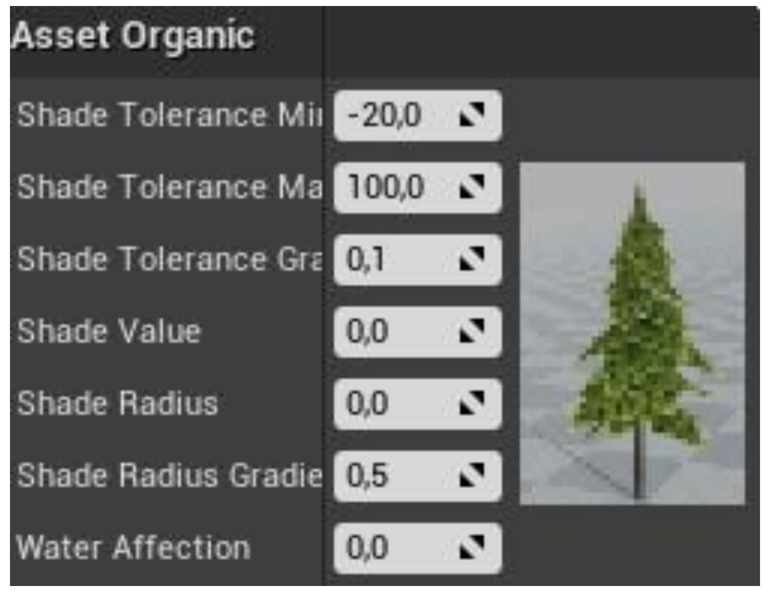
- Iterative, rule-based local-to-global model
  - Assets with bilateral placement rules
  - Asset hierarchy, assigned to biomes
  - Constrained-based placement via dart throwing

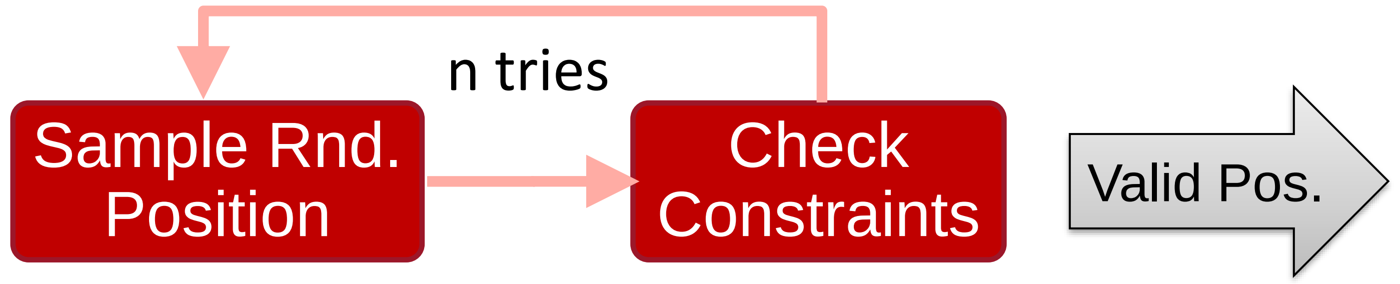


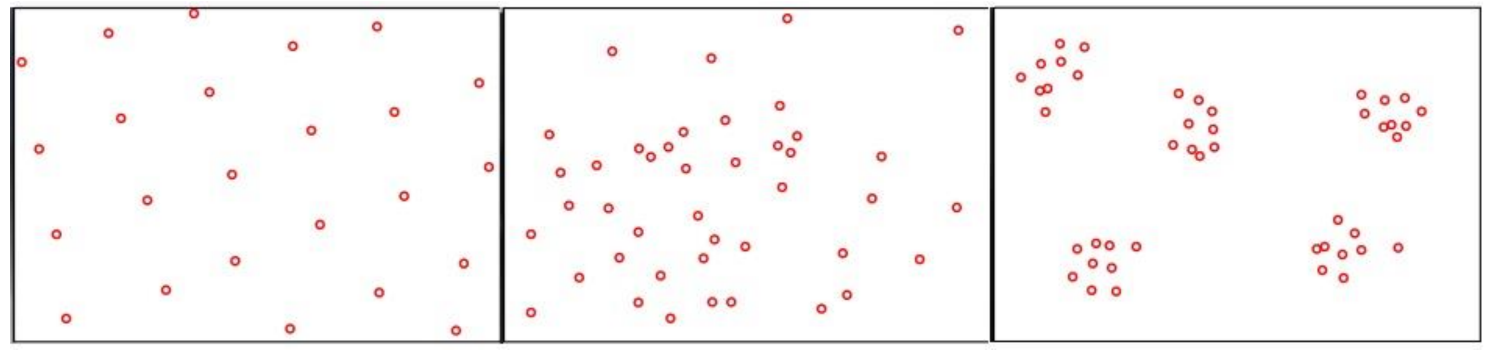


Procedural Asset Placement

- Iterative, rule-based local-to-global model
  - Assets with bilateral placement rules
  - Asset hierarchy, assigned to biomes
  - Constrained-based placement via dart throwing
- Enables emergent multi-object distributions

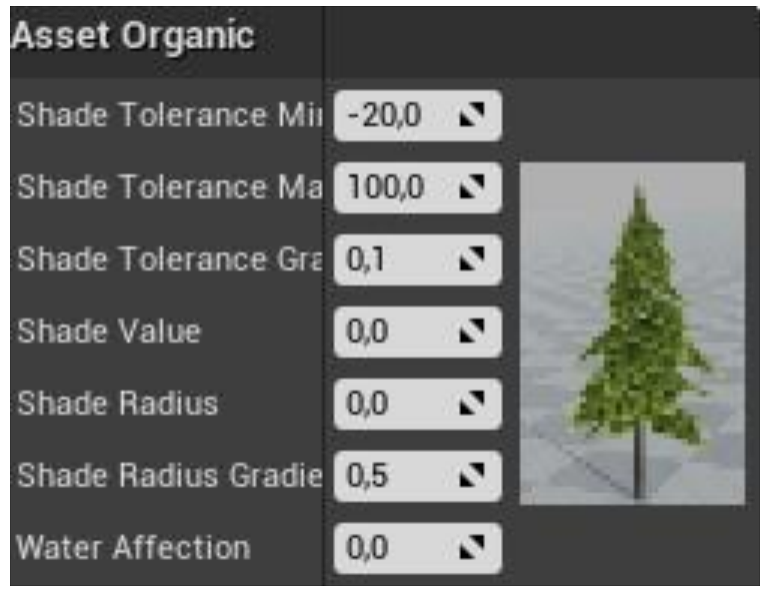


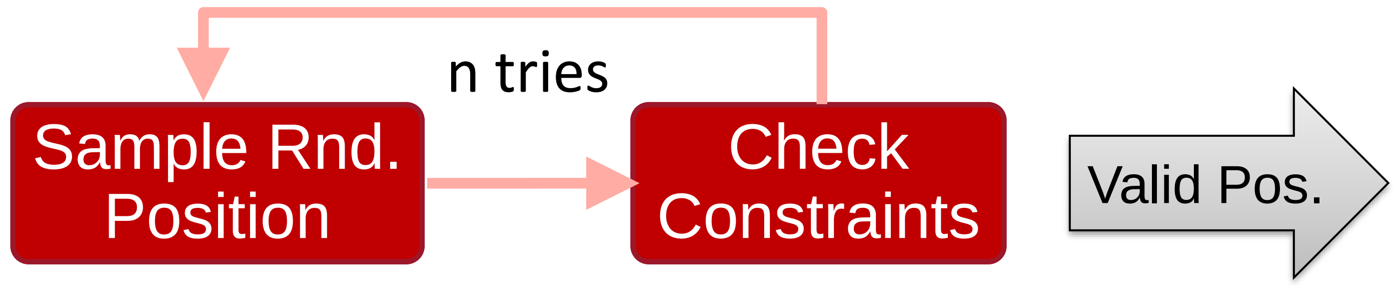


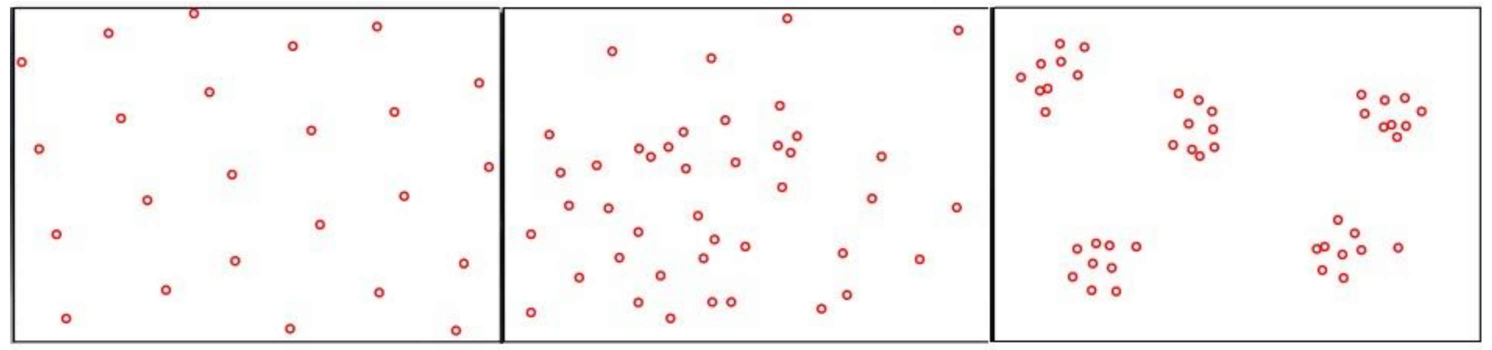


Procedural Asset Placement

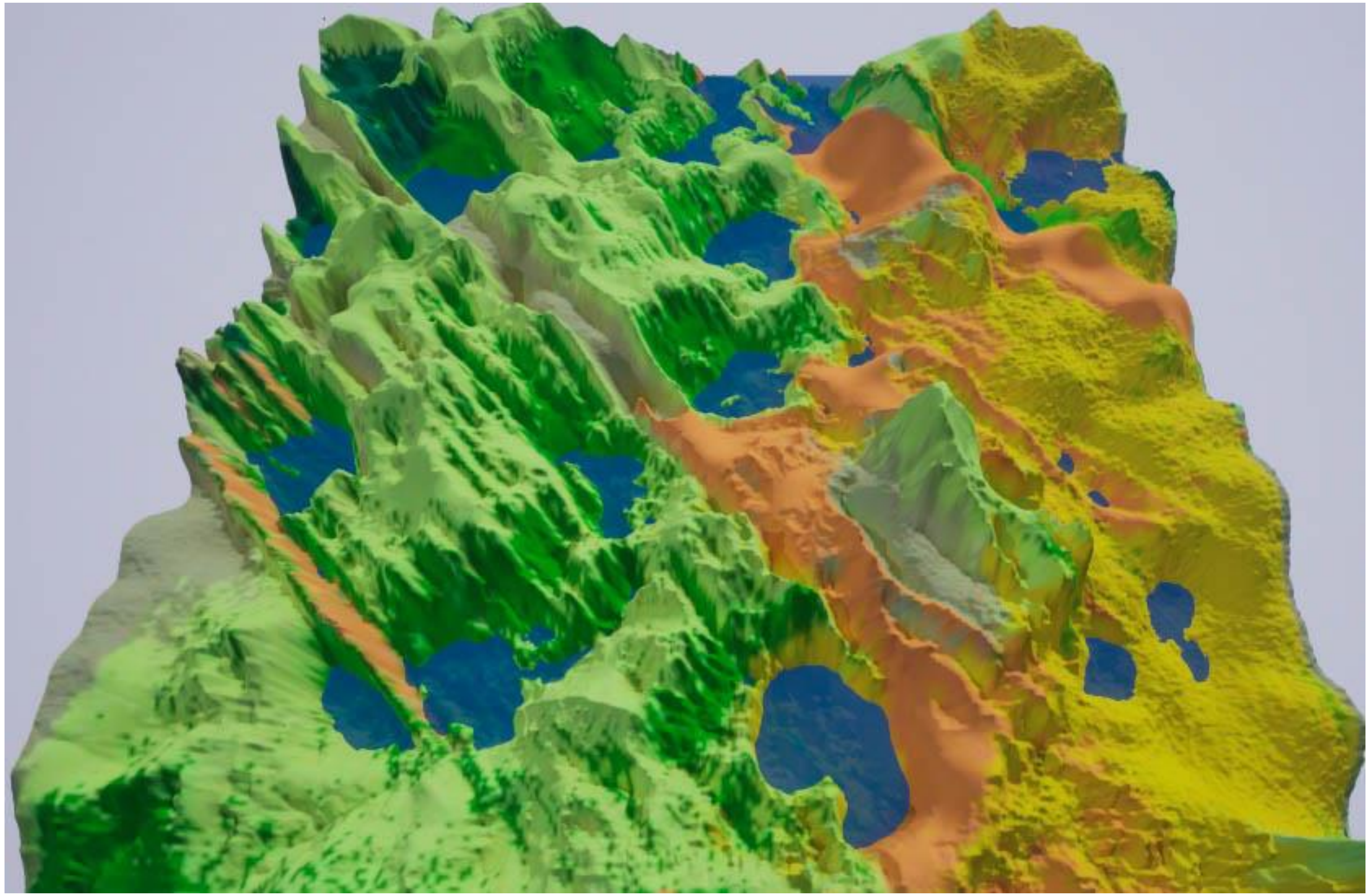
- Iterative, rule-based local-to-global model
  - Assets with bilateral placement rules
  - Asset hierarchy, assigned to biomes
  - Constrained-based placement via dart throwing
- Enables emergent multi-object distributions
- All seasons with one placement
  - Switching asset variants





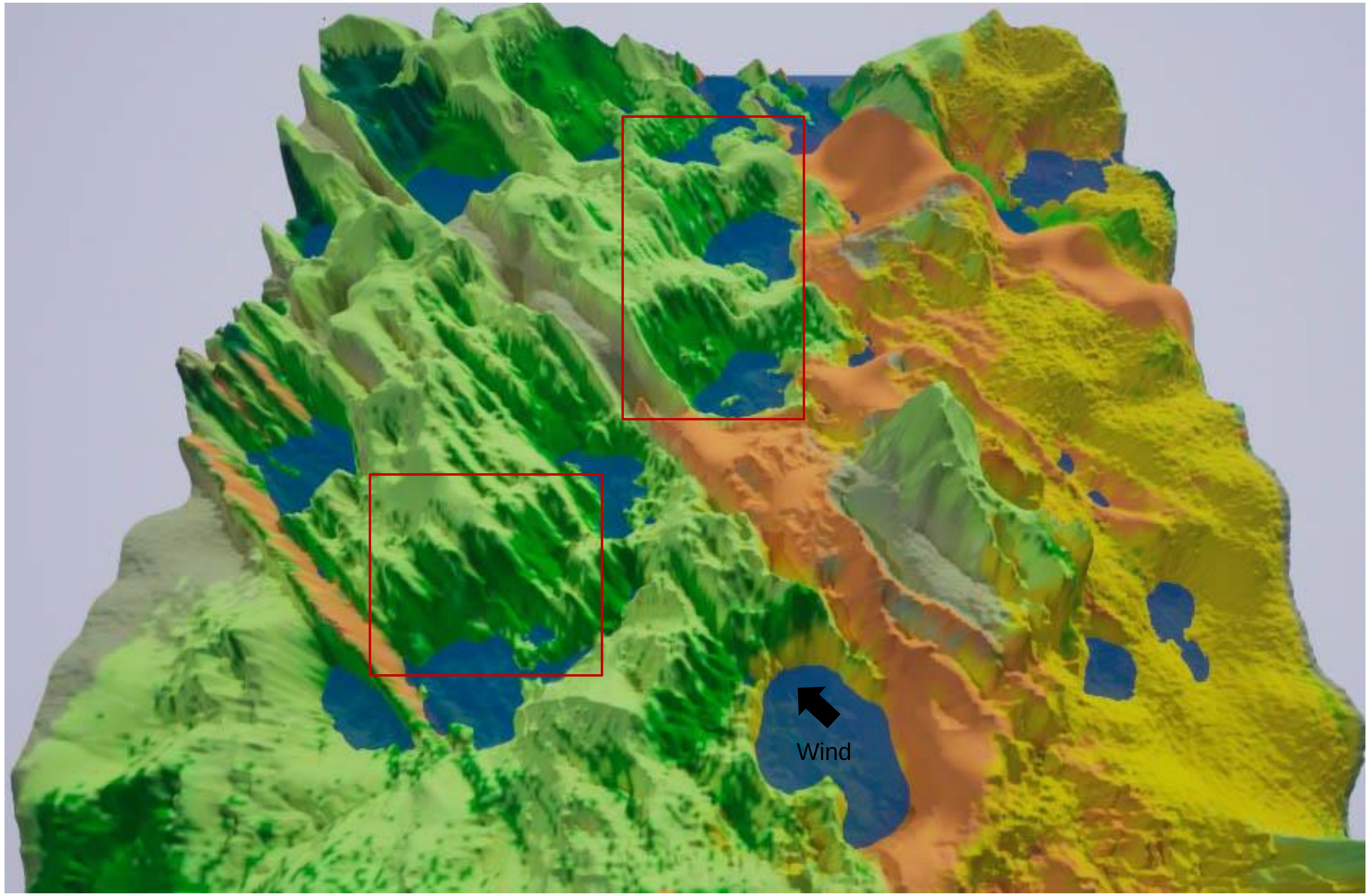


Result: Proxy View of Final Terrains



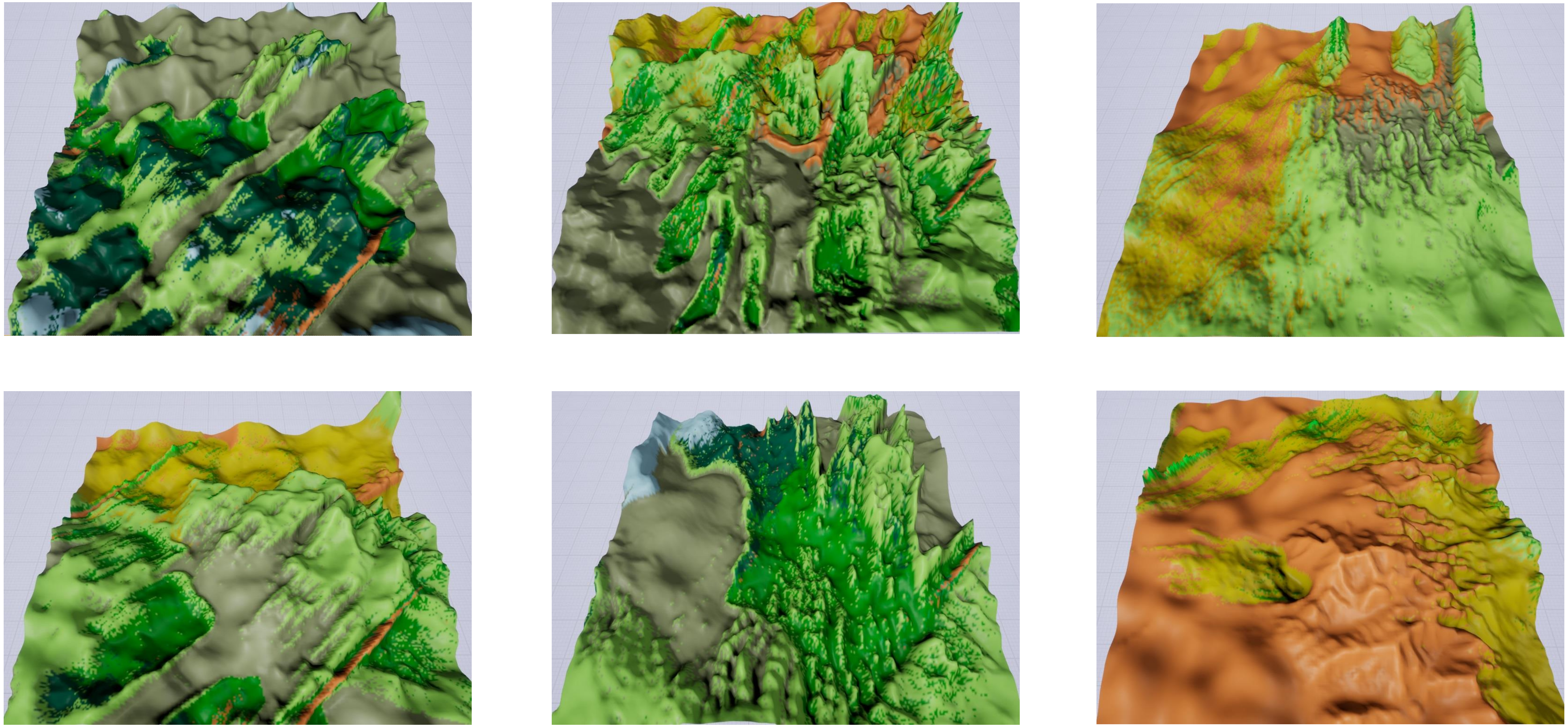
Represents $1600\ km^2$

Result: Proxy View of Final Terrains



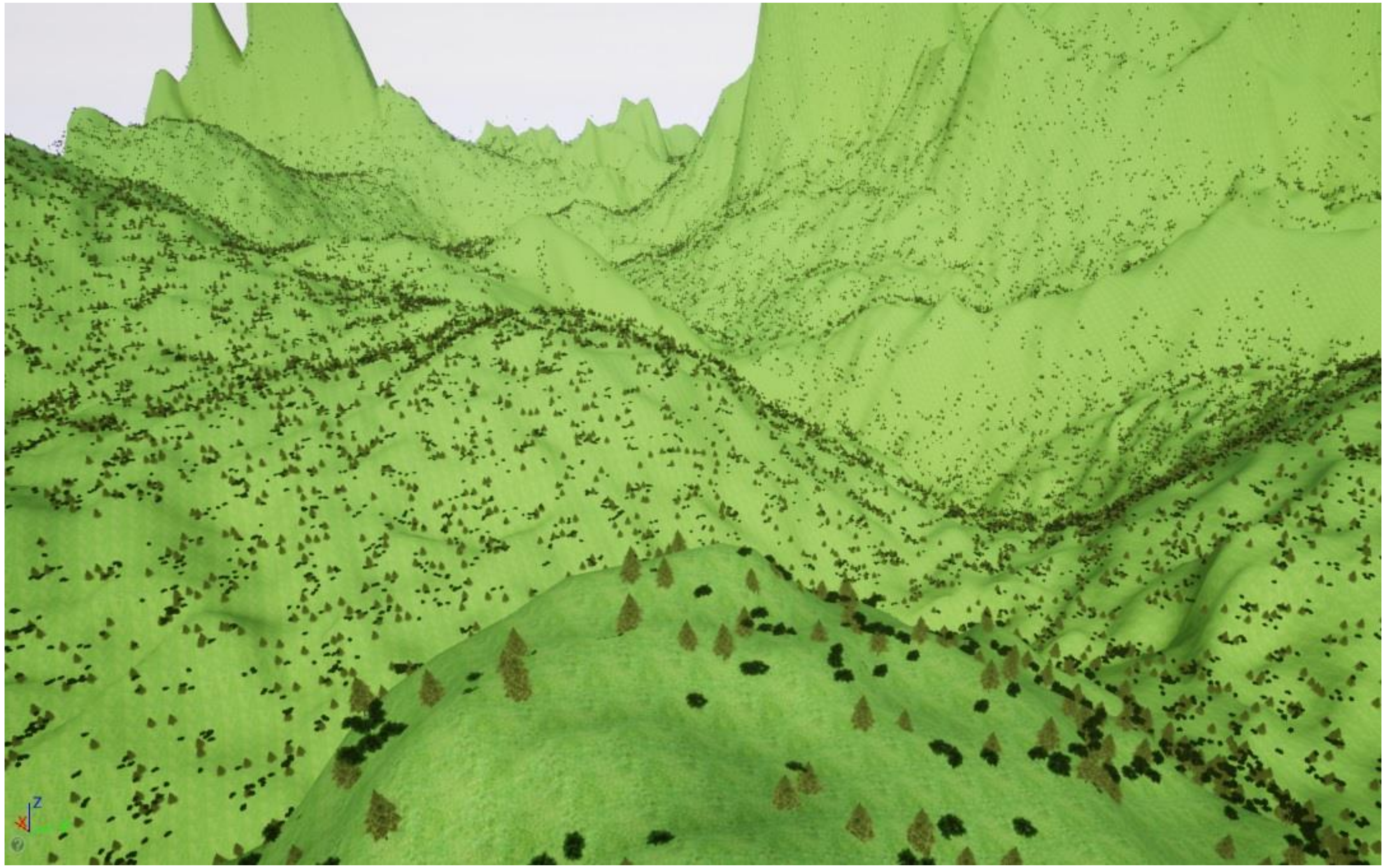
Represents 1600 km^2

Result: Proxy View of Final Terrains



Represents 1600 km^2

Result: Asset Placement



~ 200,000 instances on final terrain

Result: Asset Placement

Shrubs exclusively in shadow of dense tree clusters

Tight clusters of shrubs in open spaces between trees

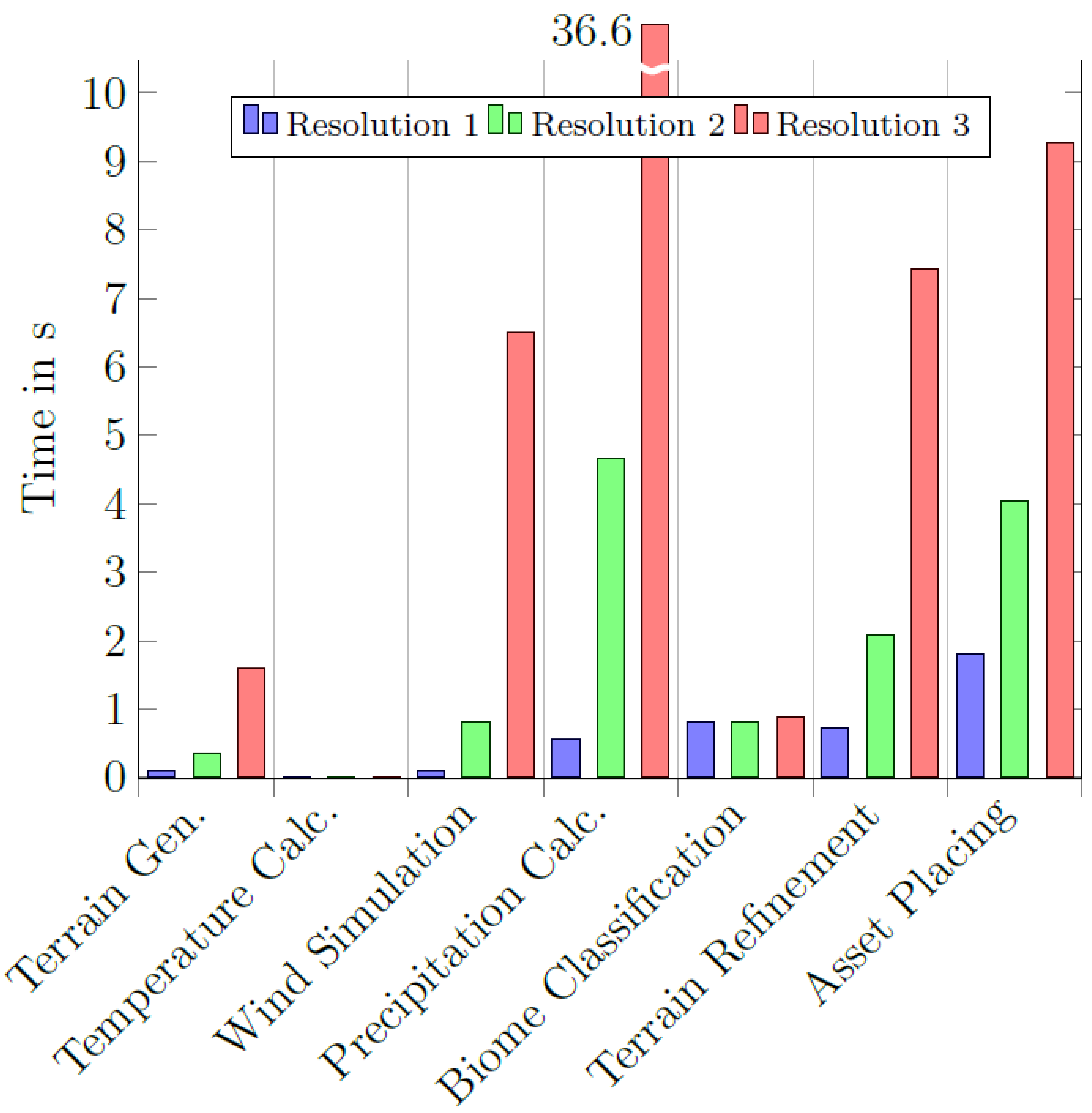
Dense, clumped shrubs around loosely grouped trees

Result: Asset Placement - Seasons

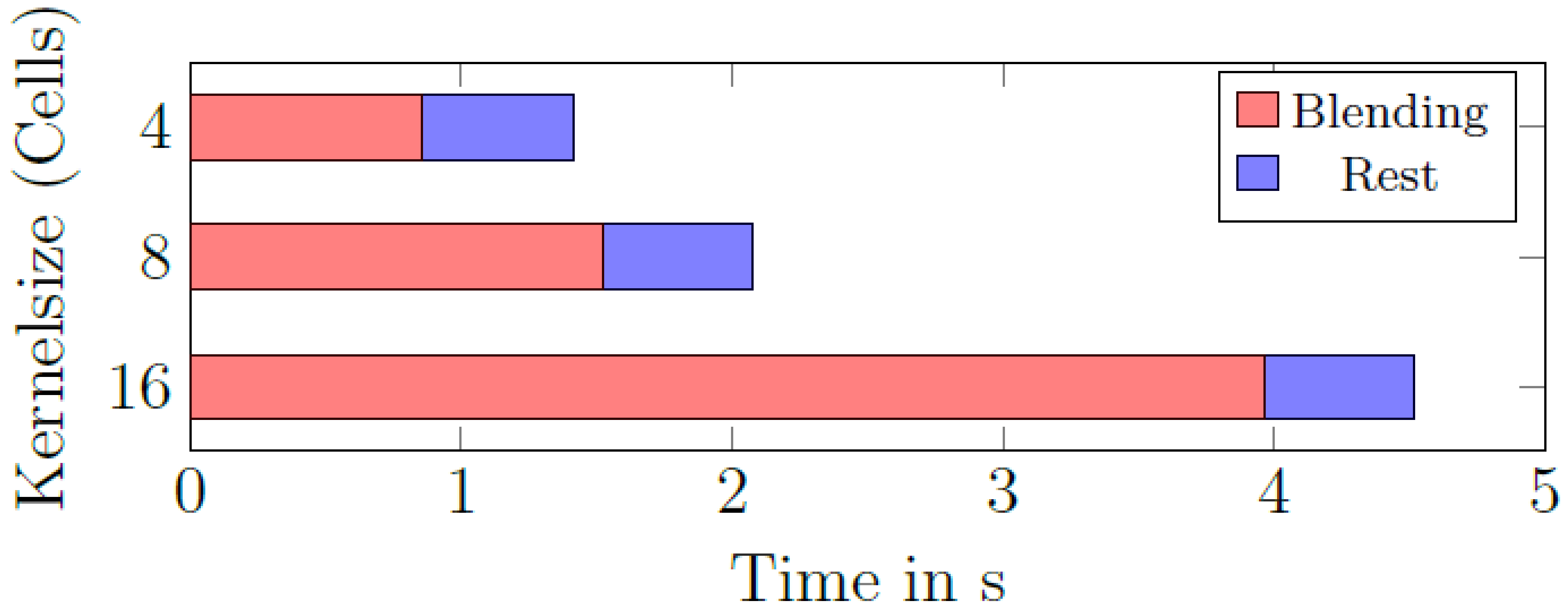
Result: Performance

Complexity:
$O(N \cdot k)$ - N: # cells
k: # iterations

Pipeline Step	Res. 1	Res. 2	Res. 3
Terrain Gen.	1024	2048	4096
Asset Placing	30	60	120
Rest	128	256	512



Result: Terrain Refinement Performance



Conclusion

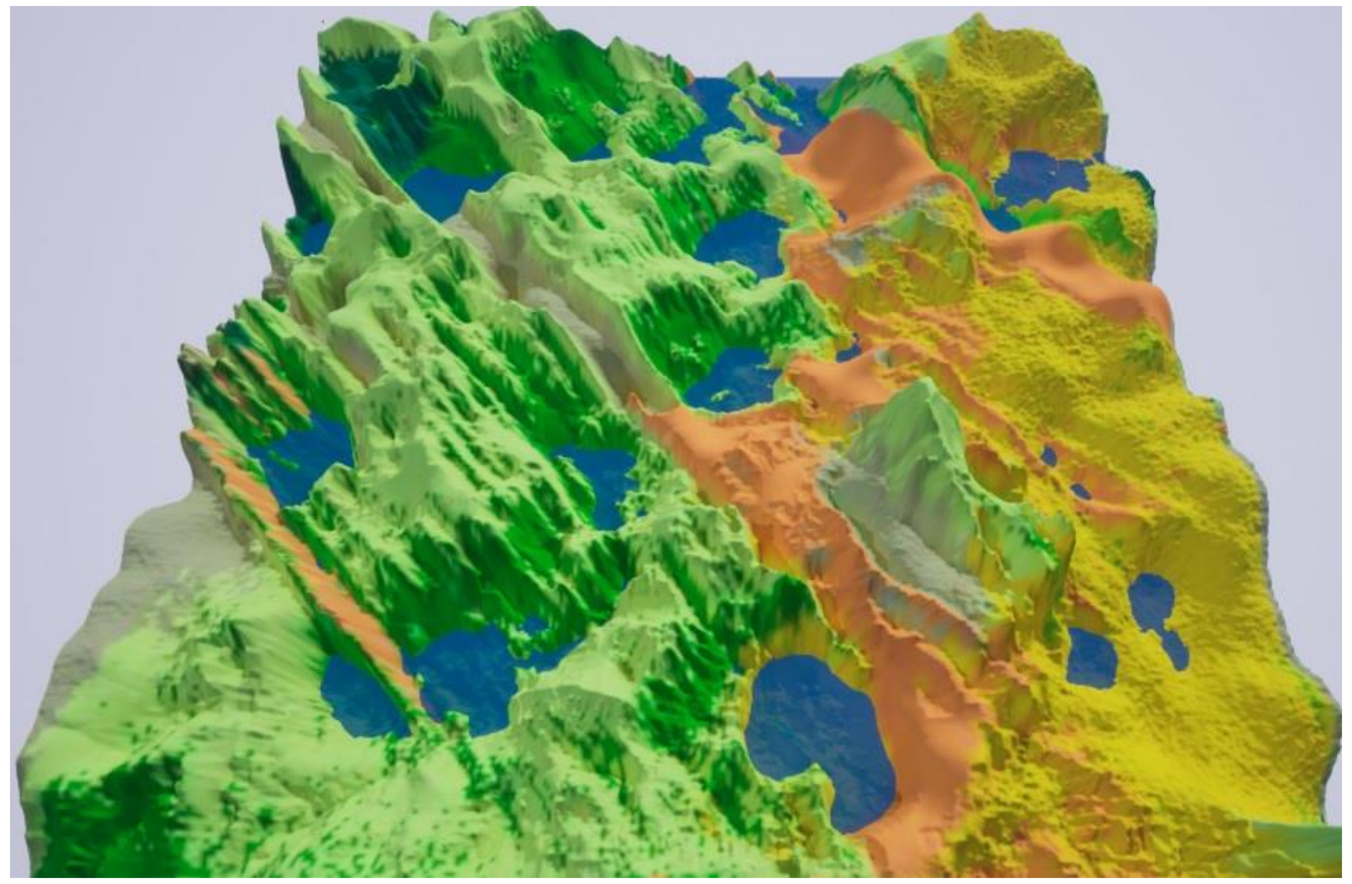
- Successful combination of synthetic, physics-based and example-based PTG

Conclusion

- Successful combination of synthetic, physics-based and example-based PTG
- Effective generation of vast, plausible-looking landscapes

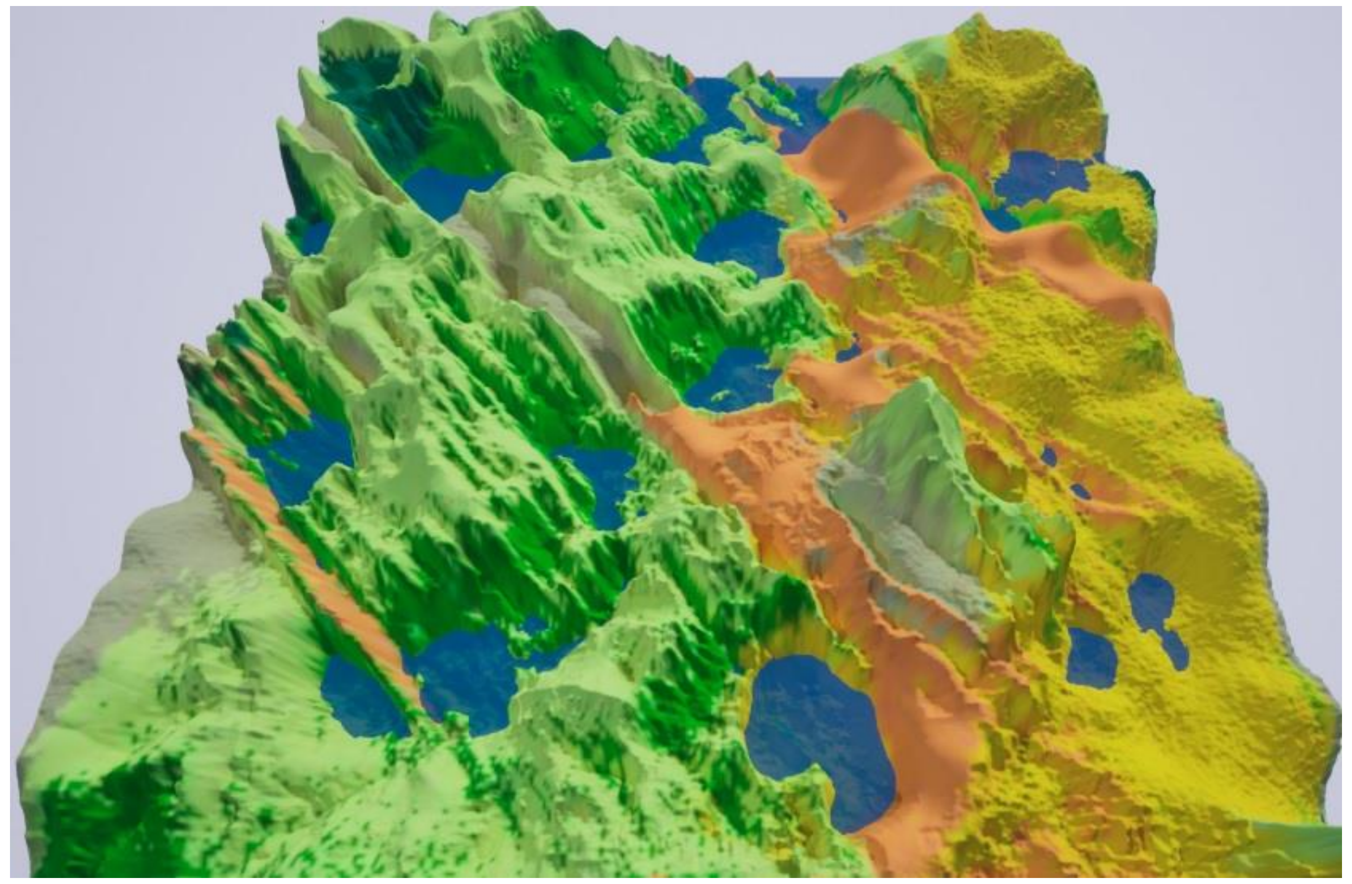
Conclusion

- Successful combination of synthetic, physics-based and example-based PTG
- Effective generation of vast, plausible-looking landscapes
- Varied landscapes as combination of biomes



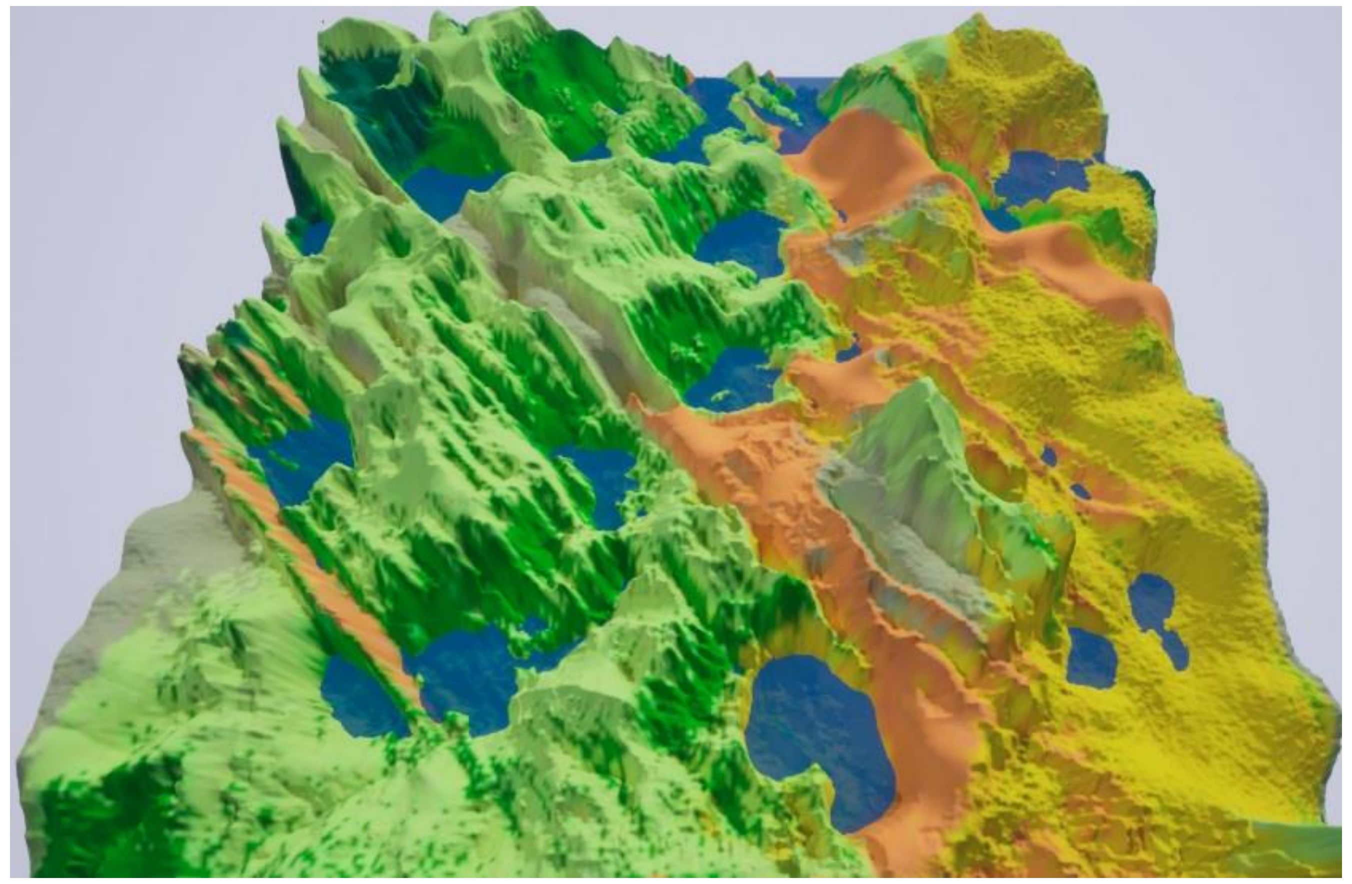
Conclusion

- Successful combination of synthetic, physics-based and example-based PTG
- Effective generation of vast, plausible-looking landscapes
- Varied landscapes as combination of biomes
- Procedural, complex rule-based asset placement



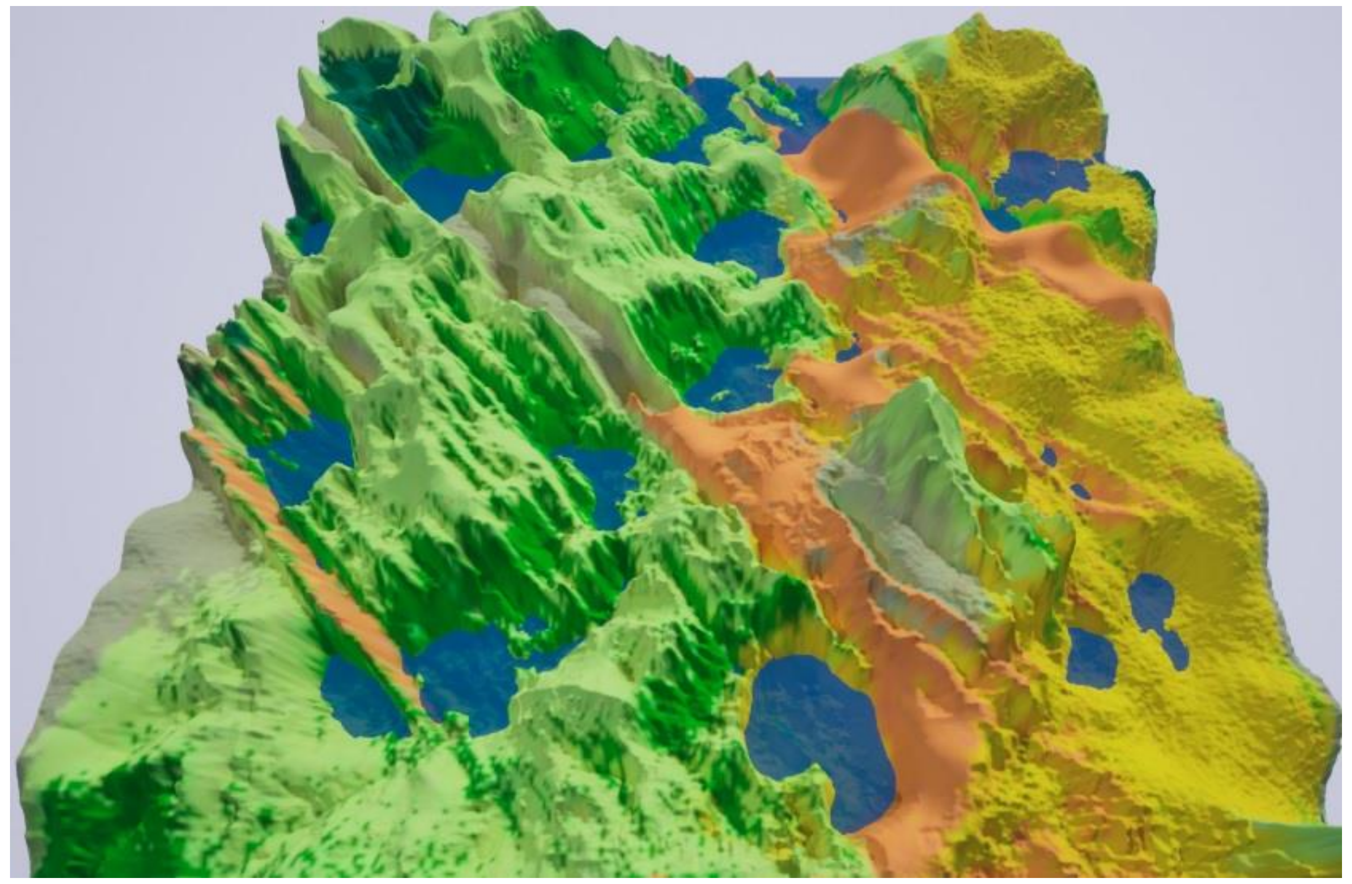
Conclusion

- Successful combination of synthetic, physics-based and example-based PTG
- Effective generation of vast, plausible-looking landscapes
- Varied landscapes as combination of biomes
- Procedural, complex rule-based asset placement
- Quick, easy-to-use iterative workflow



Conclusion

- Successful combination of synthetic, physics-based and example-based PTG
- Effective generation of vast, plausible-looking landscapes
- Varied landscapes as combination of biomes
- Procedural, complex rule-based asset placement
- Quick, easy-to-use iterative workflow
- Unreal Engine 4 integration



Future Work

- Consider geological properties and soil types

Future Work

- Consider geological properties and soil types
- Add rivers/water bodies and erosion

Future Work

- Consider geological properties and soil types
- Add rivers/water bodies and erosion
- Generate and combine DEMs using neural networks

Future Work

- Consider geological properties and soil types
- Add rivers/water bodies and erosion
- Generate and combine DEMs using neural networks
- Add sketch-based user control

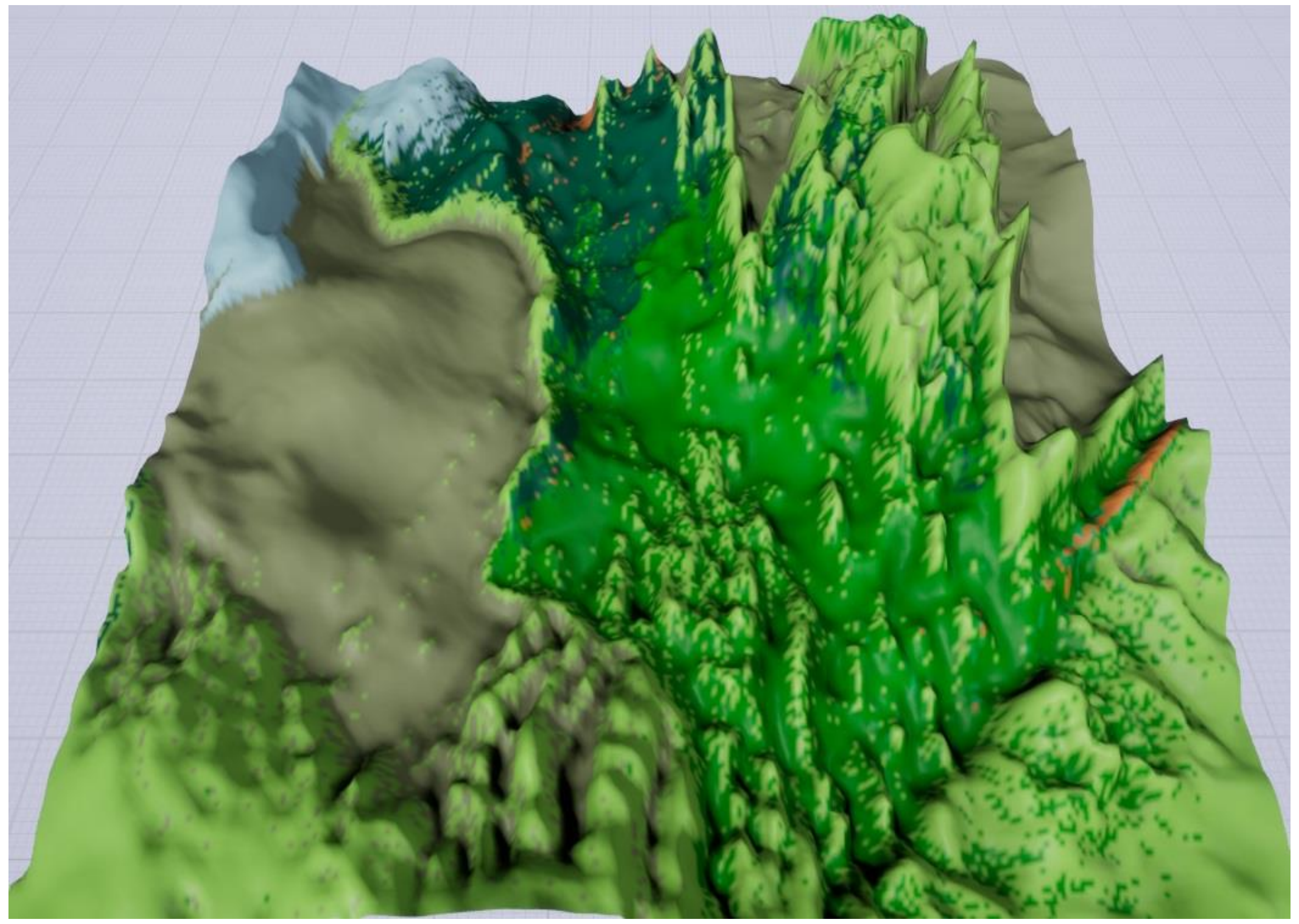
Future Work

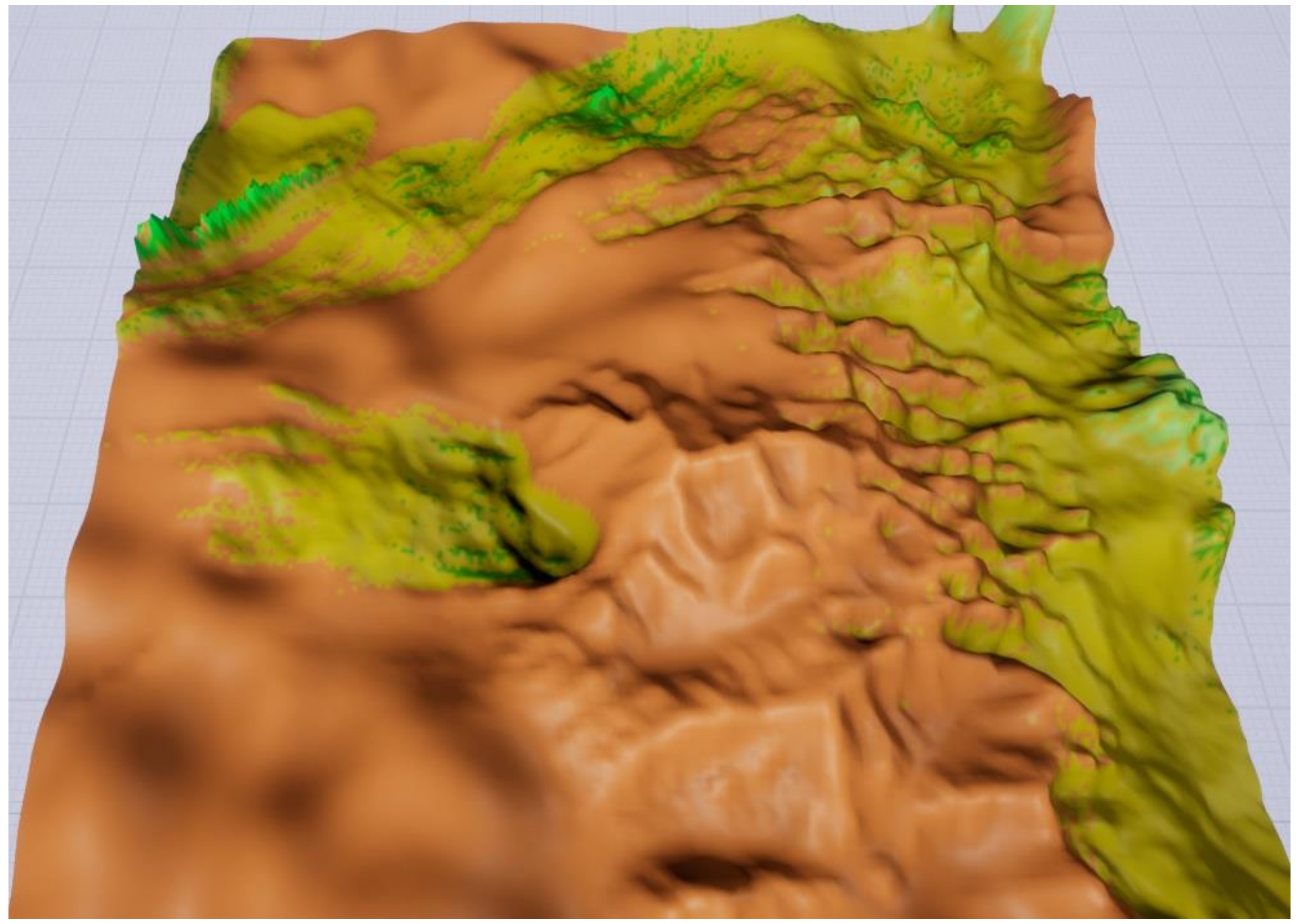
- Consider geological properties and soil types
- Add rivers/water bodies and erosion
- Generate and combine DEMs using neural networks
- Add sketch-based user control
- Improve efficiency, e.g. multi-threading

Bremen

Thank you!

Questions?





CGI2020

rfischer@cs.uni-bremen.de