

Innovazioni nella Generazione Procedurale di Terreni

Foundations of Game Design
Università degli Studi di Milano-Bicocca

Matteo Breganni 869549



Introduzione

La generazione procedurale di terreni permette di avere (per l'utente):

- **Replayability infinita**, con mondi sempre nuovi
- Risparmio di **risorse di memorizzazione**
- Supporto ai **designers**:
 - **Problemi di realismo e controllabilità dei risultati**

Paper e tecniche studiate:

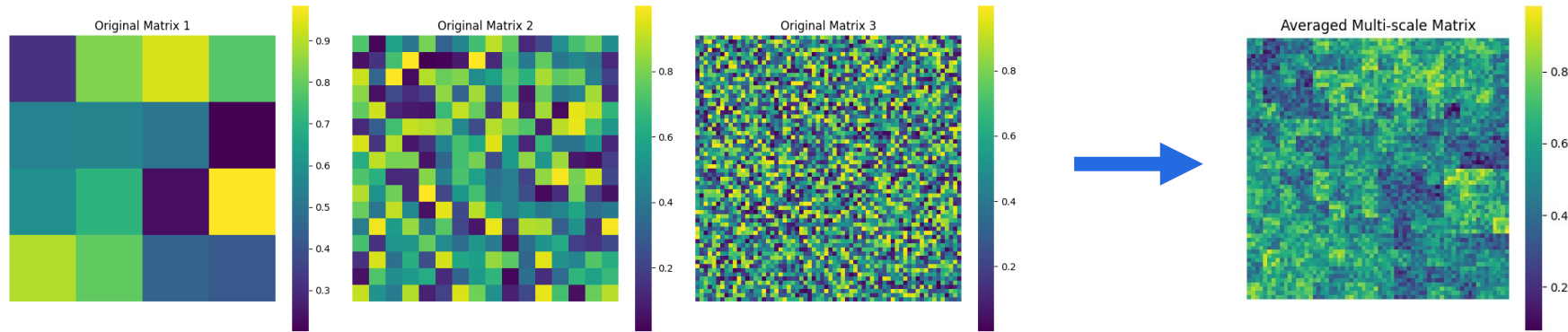
- *Procedural terrain generation with style transfer (2024)*
 - Explicit Noise
 - Perlin Noise
 - Neural Style Transfer
- *StyleTerrain: Disentangled generative model for controllable high-quality procedural terrain generation (2023)*
 - GANs
 - Disentanglement dello spazio latente con StyleGAN2
 - Super-resolution avanzata con GLEAN

Codice python e **demo** preparata per tutte le tecniche del primo paper.

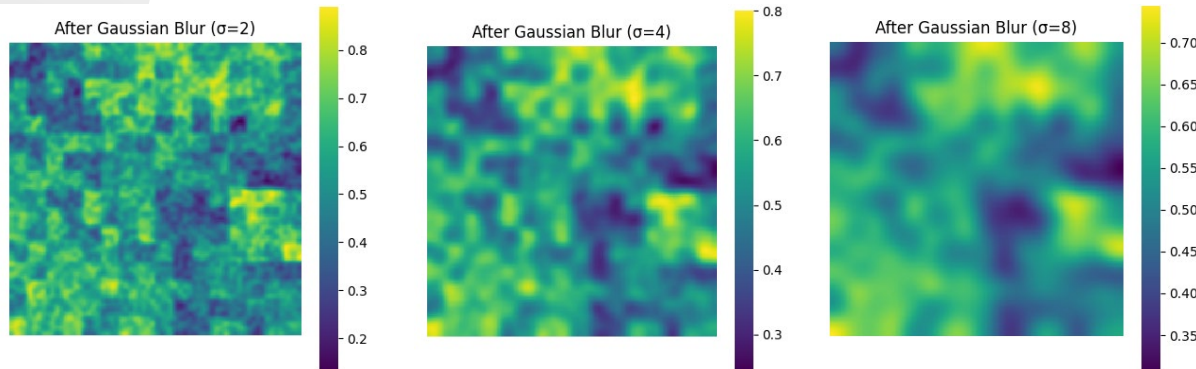
Generazione Procedurale di Terreno con Style Transfer

Explicit Noise

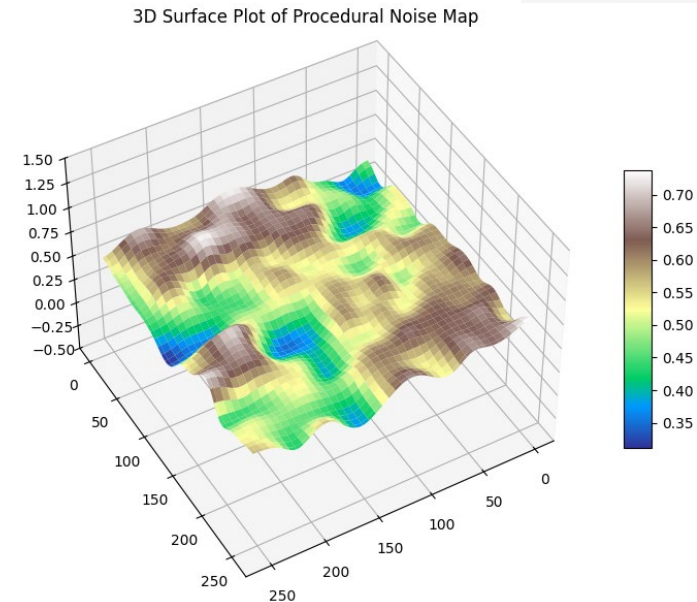
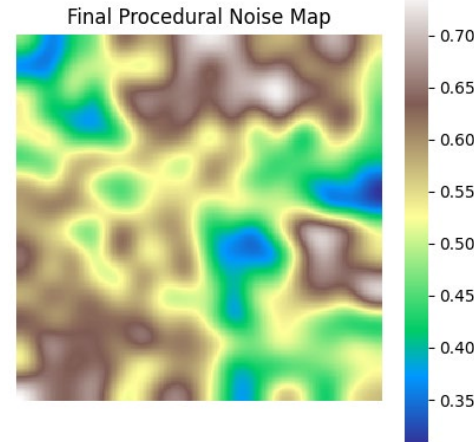
- Matrici di dimensione diversa, con **valori uniformemente distribuiti**
- Upscale, media -> valori non più uniformemente distribuiti



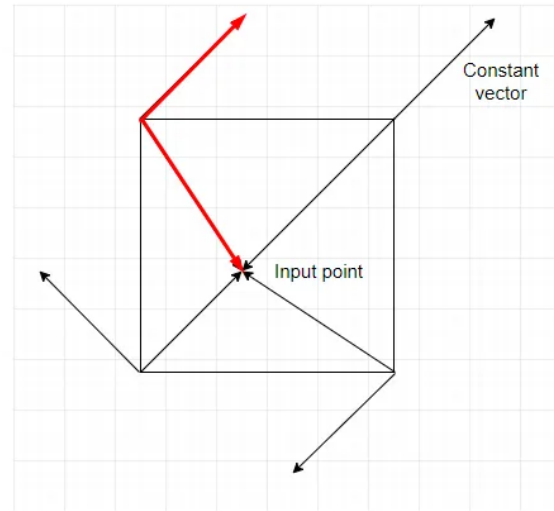
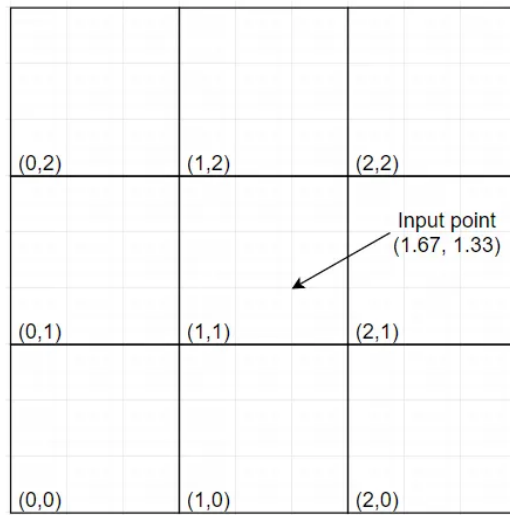
- Filtri gaussiani di dimensioni diverse



- Risultato:

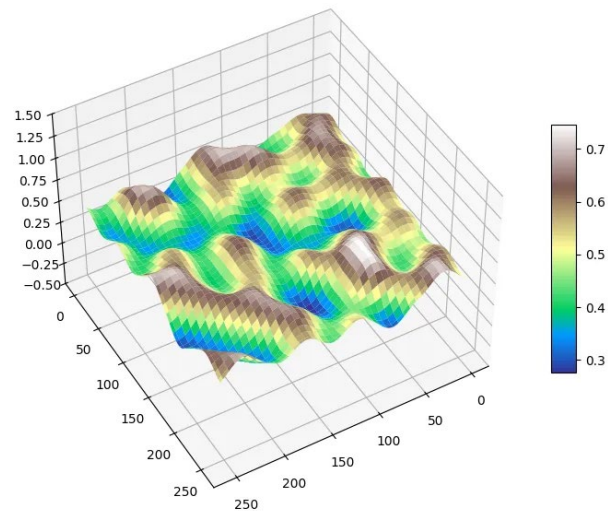


Perlin Noise

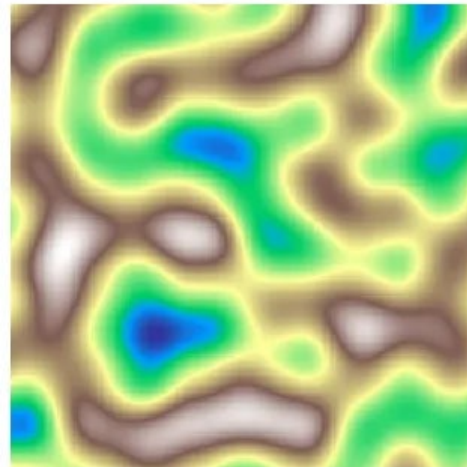


- Dot product
- Interpolazione non lineare

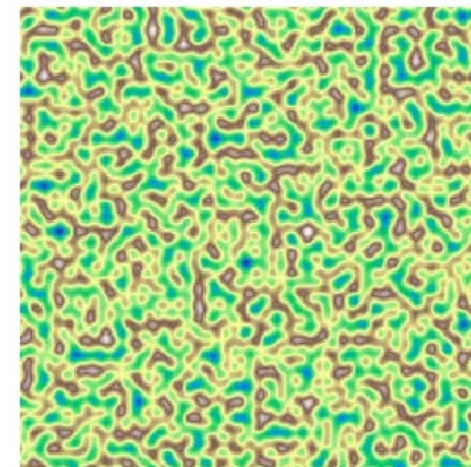
- Procedimento ripetuto per ogni punto della height map



Scale = 5



Scale = 30

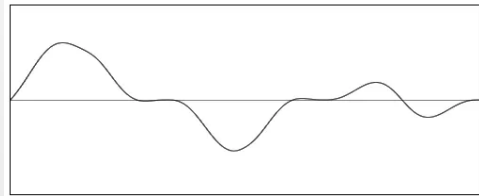


Fractal Perlin Noise

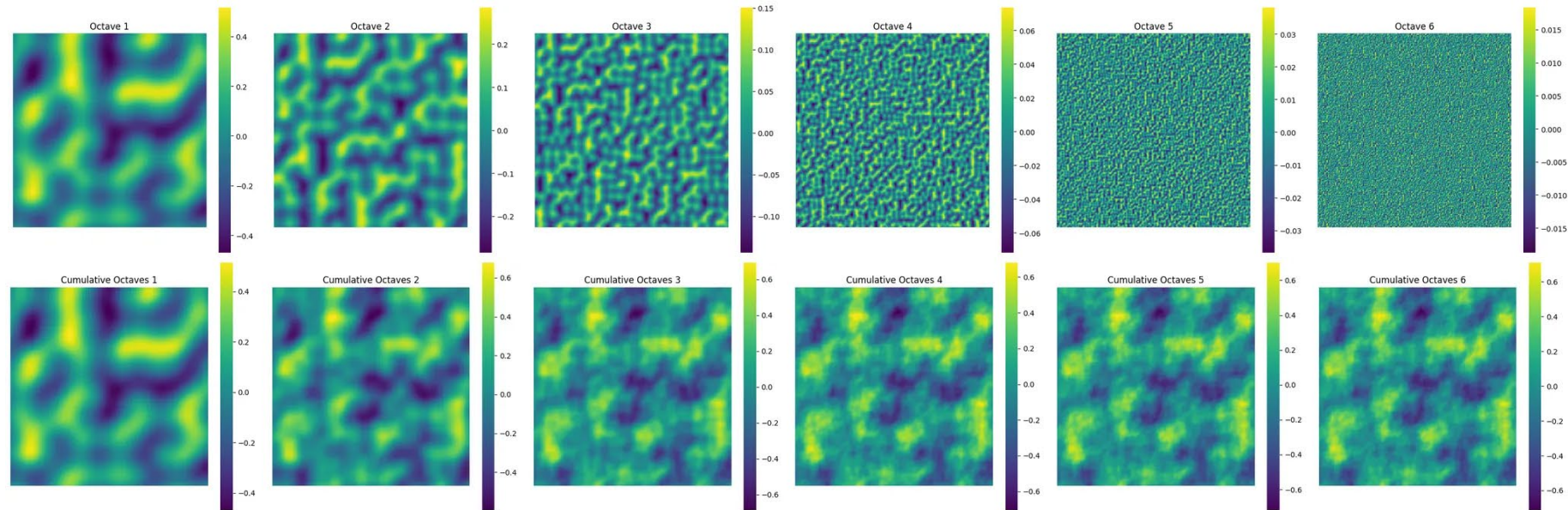
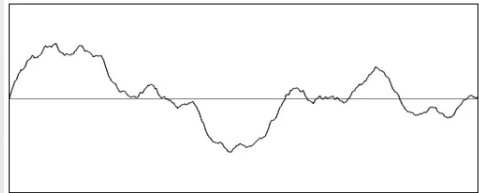
Esempio in 1D:

- octaves=6, persistence=0.5, lacunarity=2.0

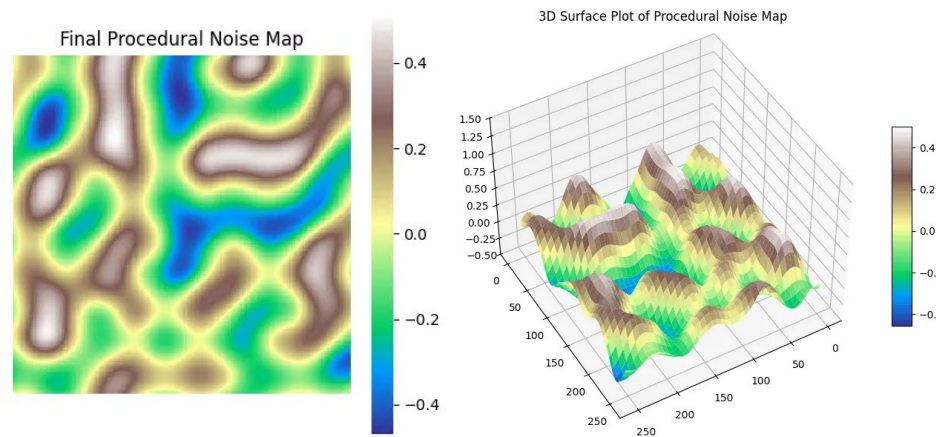
2 ottave (0.5 per la seconda):



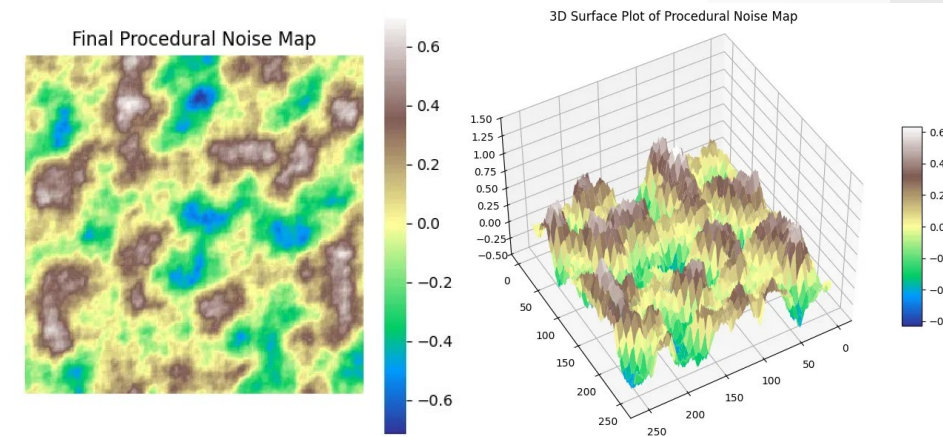
4 ottave:



Perlin Noise: una ottava



Fractal Perlin Noise: sei ottave



Style Transfer

- Applicare uno stile ad una heightmap
- Usa VGG-19 come feature extractor

- **Content Loss:**

$$L_{content} = \sum (F_{output} - F_{content})^2$$

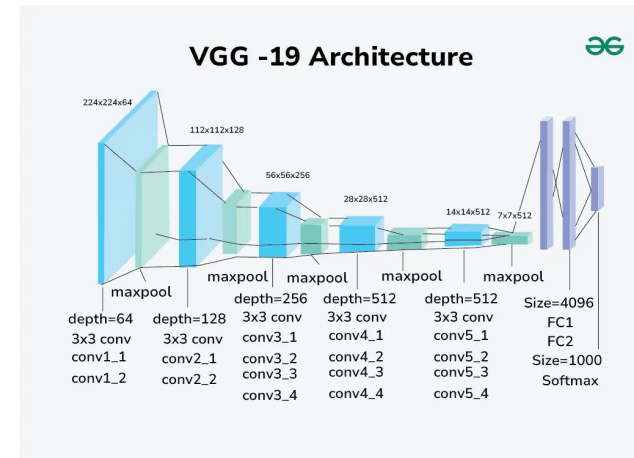
- Mean squared error su conv5_2

- **Style Loss:**

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

- **Matrici di Gram:** ogni elemento rappresenta quanto sono simili i filtri i e j (**correlazioni**).
 - Usa il primo layer convoluzionale di ciascun blocco
 - Due *Matrici di Gram* (output image e style image) per calcolare la Style Loss:

$$L_{style} = 1/4N_l^2 M_l^2 \sum_{ij} (G_{output} - G_{style})^2$$



- **Total Variation Loss:**

$$L_{TV} = \sum_{i,j} (||x_{i,j} - x_{i+1,j}|| + ||x_{i,j} - x_{i,j+1}||)$$

- Per ogni pixel dell'output image, calcola la differenza tra il pixel e quello sotto, e il pixel e quello a destra.
 - Più i pixel sono simili e più bassa sarà la loss
 - Riduce gli **artefatti**

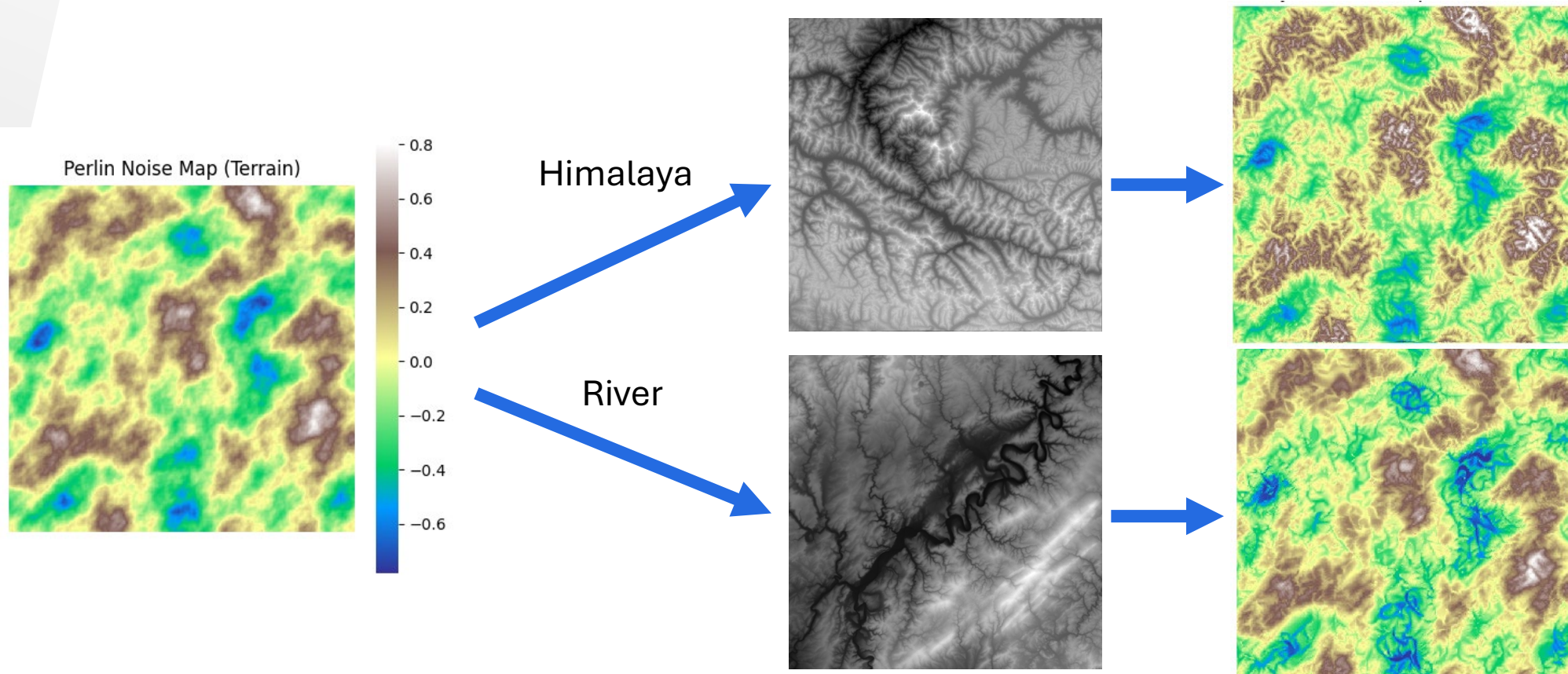
- **Final Optimization Loss:**

$$Loss = \alpha L_{content} + \beta L_{style} + \gamma L_{TV}$$

Style Transfer

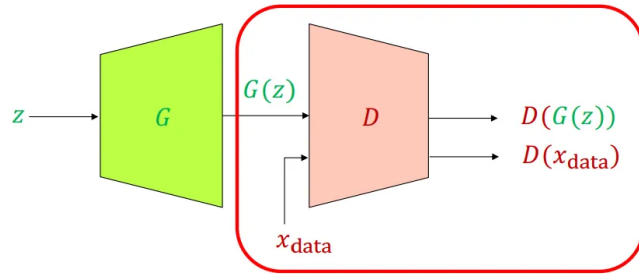
Processo di ottimizzazione:

- genera una heightmap tramite Perlin Noise o Explicit Noise (**Content Image**),
- ottieni una heightmap reale con lo stile desiderato (montagne, fiumi...) (**Style Image**),
- estrai le **attivazioni dei layers** necessari della rete neurale convoluzionale, rispetto alle due heightmap,
- esegui il **ciclo di ottimizzazione** partendo dalla **content image** e aggiornando l'immagine utilizzando la **discesa del gradiente stocastica (SGD)** con la Final Optimization Loss, con un learning rate decrescente.



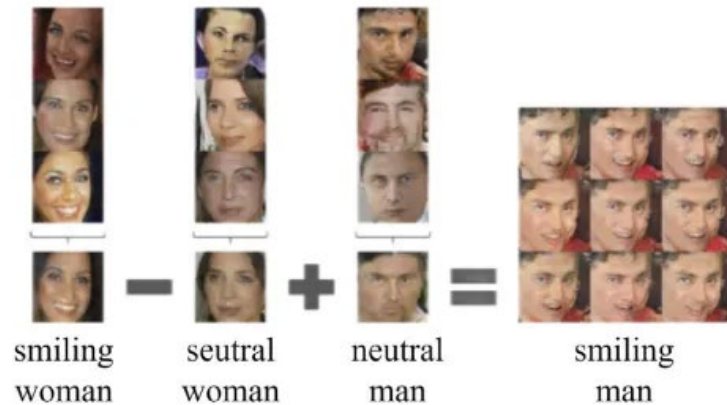
StyleTerrain: Modello Generativo Disentangled per Terreno Procedurale

Introduzione alle GAN



- Meccanismo **adversarial** per il training:
 - Generatore minimizza la loss
 - Discriminatore massimizza la loss

Il **latent code** in input z può essere manipolato, per esempio applicando operazioni aritmetiche e interpolazioni:



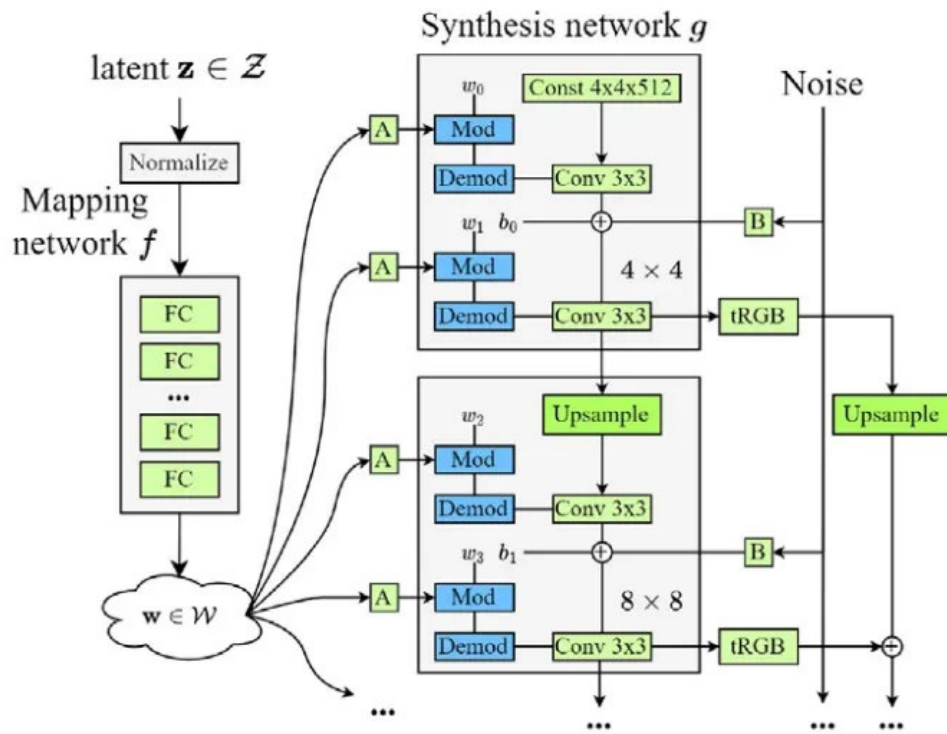
Interpolation between different points in the z space



GAN Disentanglement: StyleGAN2

Problema: poca controllabilità dell'output della GAN, per la generazione di terreni

- Spazio latente Z troppo «aggrovigliato»
- StyleTerrain adotta la rete **StyleGAN2**



(a) StyleGAN2 Generator (detailed)

Mapping Network:

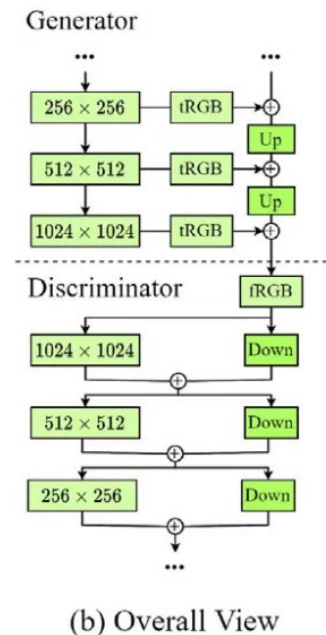
- Rete feed forward
- Trasforma lo **spazio latente** Z in W

Synthesis Network:

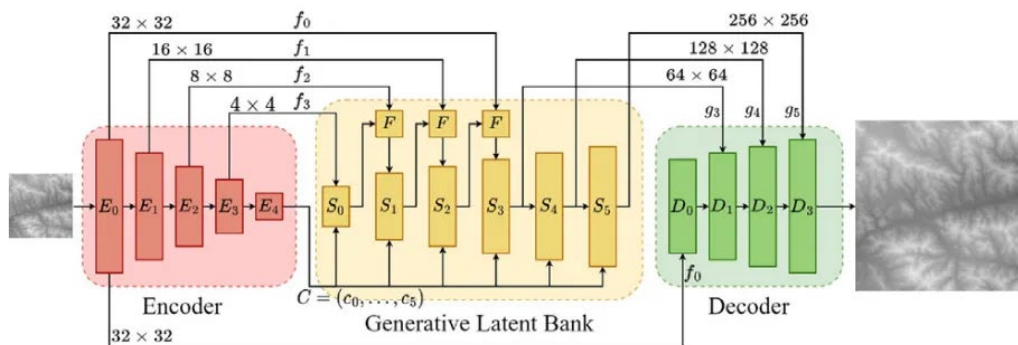
- Con l'aiuto del mapping network, impara un **mapping** che divide l'entanglement delle features
- Applica operazioni di **style transfer** a diversi livelli di risoluzione, partendo da una costante
- Ad ogni livello, applica **convoluzioni** per elaborare la feature map precedente.
- Lo spazio latente **W** modula i pesi delle convoluzioni
 - Grazie al *disentanglement* dello spazio W , ogni w_i può influenzare aspetti specifici dell'immagine
- **Noise** b_i aggiunto per introdurre variazioni e dettagli fini
- **Upsampling** interno ed esterno

Manipolazione Codice Latente

- **Livelli più bassi** (da 0 a 3) generano la struttura grossolana (layout generale)
- **Livelli intermedi** (dal 4 a 7) aggiungono dettagli come colline o fiumi
- **Livelli alti** (da 8 a 17) si occupano dei dettagli fini (rugosità, texture)
- Lo **spazio W** è fondamentale perché permette **manipolazioni** lineari che producono cambiamenti localizzati e interpretabili
- **Interpolazione** tra due codici w : transizione fluida tra due stili di terreno
- **Style mixing**: prendendo parte del codice w_1 e parte del codice w_2
- **Editing mirato**: cambiando solo alcune dimensioni di w (es: aumentare la pendenza)



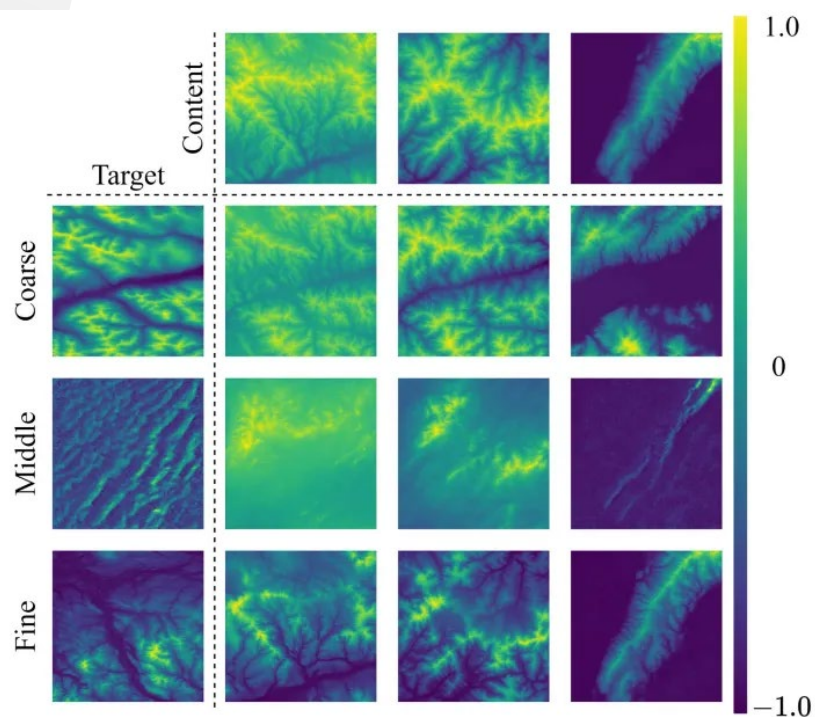
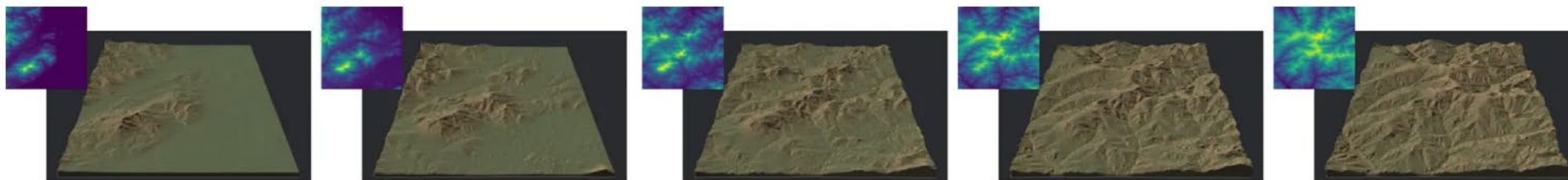
GLEAN (Generative LatEnt bANk)



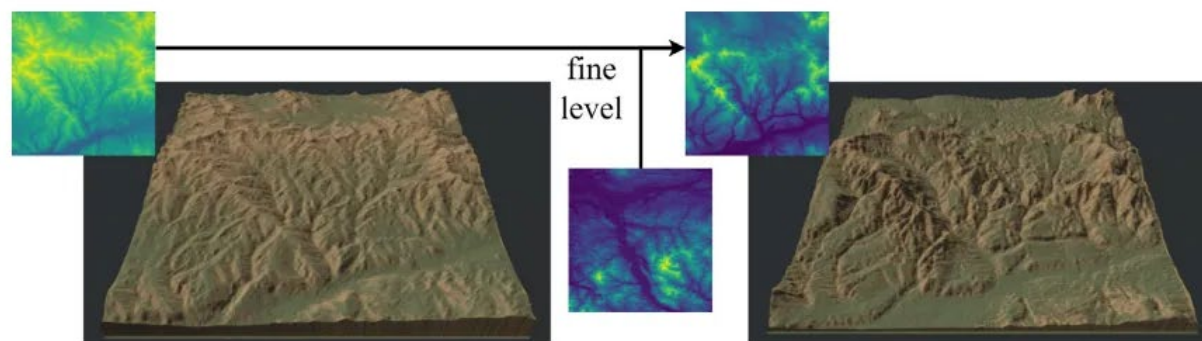
- Modello avanzato per la **super-risoluzione**
- Il **generatore** centrale usa gli stessi pesi di StyleGAN2

Risultati

- Interpolazioni:



- Style Mixing:



- Alta capacità di **controllo**
- Primo paper a introdurre il disentanglement per la generazione di heightmap di terreni

Grazie per l'attenzione

Innovazioni nella Generazione Procedurale di Terreni

Matteo Breganni 869549

