Mycelium 3D

Grafica 3D con Context Free Art

www.fdsa.it/mycelium3d

Mycelium 3D 2014

Immagine in prima pagina realizzata con Context Free Art e Mycelium 3D.

Requisiti ed uso della libreria

Mycelium 3D è scaricabile all'indirizzo http://www.fdsa.it/mycelium3d

Per utilizzare questa libreria è necessario Context Free Art v. 3. Tutte le funzioni che riguardano la gestione dei colori e del canale alfa che si utilizzano sono quelle messe a disposizione nativamente da CFA. Occorre, naturalmente, un po' di dimestichezza con CFA.

E' sufficiente includerla mediante il comando import. Il file mycelium3d.cfdg si deve trovare nella stessa directory del file di disegno che stiamo facendo. Prima del comando di importazione è necessario specificare il livello di pseudo-ombreggiatura attraverso la costante shade.

```
shade = .5
import Mycelium3D_2014.cfdg
```

Concetti di base

Riferimento 3D locale

Il riferimento 3D locale è un vettore (una lista di 18 numeri) che contiene tutte le informazioni che specificano:

- la posizione nello spazio 3D.
- l'orientamento dei 3 assi ortogonali.
- la scala dei 3 assi.
- proprietà pseudo-fisiche come peso, rigidezza, età.

È visualizzabile come un cubo ruotato e deformato, oppure come tre assi coordinati ruotati e scalati nello spazio. Quasi tutte le operazioni che compiamo riguardano la creazione, lo spostamento, la rotazione la scala, la duplicazione di riferimenti 3D locali.

Perché locale? Riferimento locale è contrapposto a riferimento assoluto. Nel senso che una volta che abbiamo modificato il riferimento le successive modifiche avvengono sempre rispetto agli assi cartesiani del riferimento stesso (riferimento locale) e non rispetto alle coordinate assolute dello spazio 3D.

Muovere l'oggetto è come pilotarlo da un punto di vista soggettivo.
C'è analogia con quanto avviene normalmente nell'uso di Context free art in 2D.

Spazio 3D, 2D, telecamera, primitive

Lo spazio in cui "vivono" gli oggetti creati con M3D ha 3 dimensioni. Per la loro rappresentazione (su una canvas 2D) è necessario utilizzare una telecamera con cui inquadrarli per la visualizzazione sullo schermo.

CANVAS 2D

E' lo spazio bidimensionale su cui appare il disegno a tutti gli effetti - l'area di disegno sullo schermo.

FORME PRIMITIVE

Le forme primitive di base sono poche e semplici: sprites (cerchi), cubi linee e tubi.
Lo scopo della libreria è infatti creare disegni molto complessi formati anche da milioni di elementi che devono essere di veloce calcolo e rappresentazione.
Nel caso servisse, creare nuove primitive è comunque molto semplice, il codice delle primitive implementate può essere di agevole guida.

Operazioni con riferimenti 3D

Creazione ex novo e duplicazione

Un riferimento 3D si crea con la funzione Sys()

```
W = Sys()
```

W è inizialmente un riferimento unitario allineato con gli assi principali.

per duplicare *W* è sufficiente usare l'operatore =

```
Duplicato =  W
W1 = W
```

Spostamento

```
W = MoveX( 1, W )  // sposta W di  1 lungo il suo asse X
W = MoveY(-1, W )  // sposta W di -1 lungo il suo asse Y
W = MoveZ(.5, W )  // sposta W di .5 lungo il suo asse Z

W = MoveXYZ(mx, my, mz , W )  // funzione di spostamento generica
```

Rotazione

```
REF = RotX( 10, W )  // ruota W di  10° attorno al suo asse X
REF = RotY(  4, W )  // ruota W di   4° attorno al suo asse Y
REF = RotZ(-18, W )  // ruota W di -18° attorno al suo asse Z
```

Scala e simmetria

```
W = Scale (.9, W )     // scala in modo omogeneo nelle tre dimensioni

W = ScaleX(.5, W )    // scala solo l'asse X
W = ScaleY(.7, W )    // scala solo l'asse Y
W = ScaleZ(.5, W )    // scala solo l'asse Z

W = ScaleX(-1, W )    // la scala può essere negativa: in questo caso il
                      // riferimento sarà trasformato simmetrico
                      // rispetto al piano YZ
```

Telecamera

Preimpostate

Sono preimpostate alcune telecamere, i loro nomi dicono già tutto:

- FRONT
- BACK
- TOP
- BOTTOM
- LEFT
- RIGHT

CAM equivale a FRONT

Camera personalizzata

Tuttavia è facile creare una telecamera personalizzata con la funzione Camera:

```
CAM = Camera( rotazioneY, rotazioneX, distanza )
```

La telecamera osserva di default il punto di coordinate (0,0,0)
Per traslare la telecamera (e il punto centrale osservato) si usa la funzione *MoveCamera*:

```
CAM = MoveCamera( mx, my, mz, CAM )
```

Le coordinate *mx, my, mz* sono assolute.

Oggetti 3D primitivi

Gli oggetti 3D primitivi sono shape con 2 o 3 parametri che specificano sempre almeno un sistema di riferimento e la telecamera da usare.
E' possibile specificare il colore nei normali parametri di CFA, (vanno comunque sempre usate le parentesi quadre finali).

Lo sprite

Lo sprite è l'elemento 3D più semplice a disposizione. Esso è un cerchio disegnato sempre perpendicolare rispetto all'osservatore. (Si visualizza sempre come un cerchio scalato).
Per disegnare uno sprite occorre usare un sistema di riferimento (W) che specifica il centro e la scala del cerchio e la telecamera da cui lo osserviamo (CAM)

```
SPRITE(spritesize, W , CAM)[ b 1 h 120 sat .5 a -.9 ]
```

Faccia Quadrata

Quadrato ruotato nello spazio, orientato secondo gli assi dl sistema di riferimento, perpendicolare a Y. Per disegnarlo occorre sempre specificare un riferimento 3D e una telecamera:

```
FACE(cubesize, W , CAM)[ ]
```

Il cubo

Detto anche esaedro… Per disegnarlo occorre sempre specificare un riferimento 3D e una telecamera:

```
CUBE(cubesize, W , CAM)[ ]
```

Linee, nastri e tubi

Se si disegnano elementi allungati probabilmente linee, nastri o tubi possono dare risultati migliori di sprite e cubo. Per disegnare linee e tubi occorre specificare non uno ma 2 riferimenti (*W1*, *W2*) che corrispondono alle 2 estremità del segmento o dello spezzone di tubo.
Solitamente, duplicheremo il riferimento prima di modificarlo. Il duplicato non modificato lo useremo successivamente come prima estremità della linea o il tubo. L'elemento modificato è usato come l'altra estremità.

Nota: quando si usano i nastri o i tubi gli spostamenti devono avvenire lungo l'asse Y.

```
TRACE(pensize, W1, W2, CAM)[ ]  // spessore pensize indipendente dalla scala.
MARKER(pensize, W1, W2, CAM)[ ] // penna con tratto pennarello.
LINE(pensize, W1, W2, CAM)[ ]   // penna con spessore variabile proporzionale
                                // alla scala dei riferimenti.
TAPE(pensize, W1, W2, CAM)[ ]   // nastro che segue spostamenti in Y.
TUBE(pensize, W1, W2, CAM)[ ]   // tubo adatto ad essere usato
                                // con spostamenti lungo Y.
```

W1 e W2 sono i 2 sistemi di riferimento che specificano le 2 estremità di linea o tubo.
CAM deve essere una telecamera.

Controllo ricorsività

Terminare l’esecuzione di shape ricorsive

In molti casi vorremo usare shapes ricorsive così come facciamo normalmente con CFA in 2D.
Non potendoci avvalere del modo automatico ci affidiamo a 2 funzioni da associare a [s]:

PRIMO MODO: FUNZIONE MINSIZE()

```
RecursiveShape(W)[ s MinSize( 0.1 , W) ]
```

Quando invochiamo RecursiveShape, essa verrà terminata se W ha dimensioni inferiori a 0.1

SECONDO MODO: FUNZIONE MAXAGE()

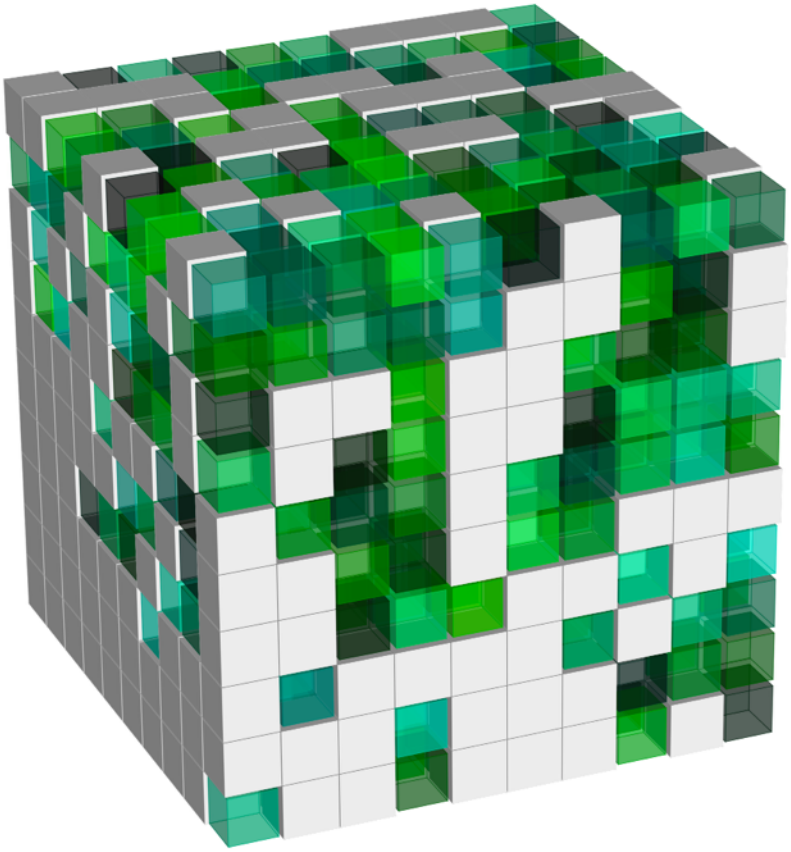
```
RecursiveShape(W)[ s MaxAge( 10 , W) ]
```

RecursiveShape, termina automaticamente se *age >= 10*; *age* è una variabile del sistema di riferimento *W* che è incrementata automaticamente ogni volta che usiamo la funzione *Move*.
E’ perciò collegata al numero di iterazioni della shape ricorsiva.

Esempi

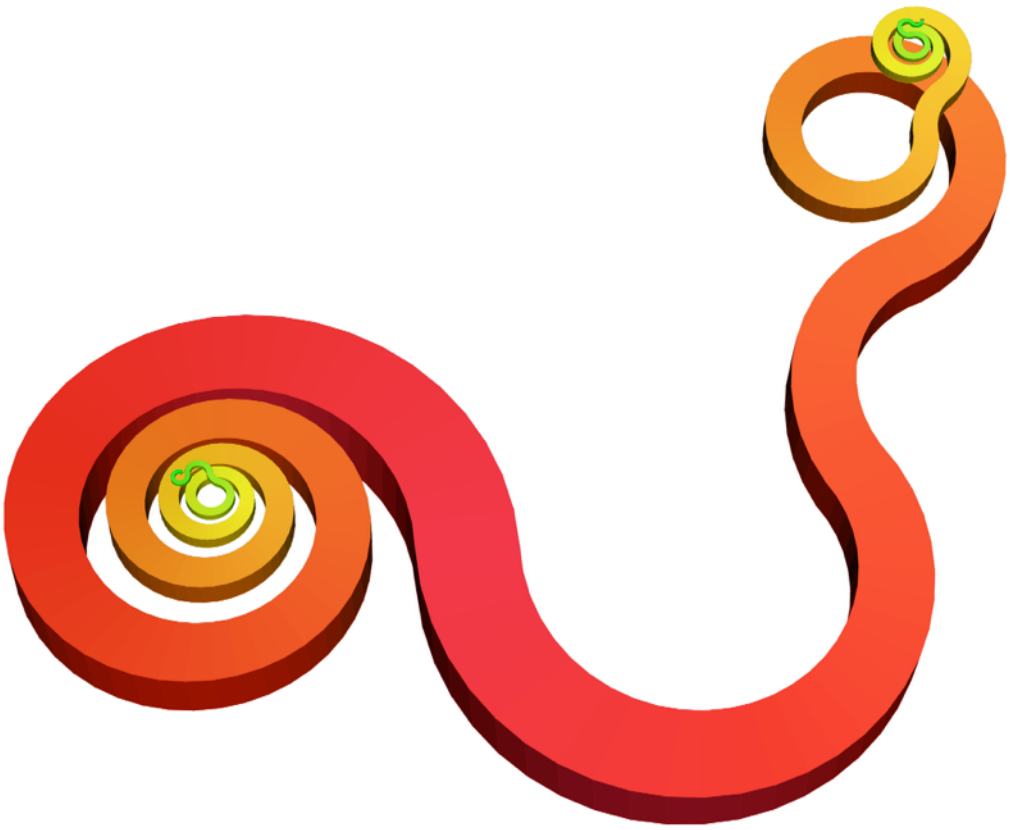
Sfera di sprites

```
shade = .45
import Mycelium3D_2014.cfdg

CF::Background = [b 1 ]
CF::Size = [s 60]

K=Camera(0, 90, 400)  // define the camera

startshape Sphere(180, Sys()) // here we create the original Reference System

shape Sphere (count, vector18 W)  // remember to specify vector18
{
       SPRITE(36.5, W, K)[a -.2]

       loop i = count/1  [ h phi  sat .03 ]
       {
              kk = asin(-1 + 2*i/count)
              W1=RotX( (phi*i) , W)
              W1=RotZ( (kk) , W1)
              W1 = MoveY(20 , W1)
              SPRITE(3.5, W1, K)[ b 1  a -.03   ]  // Colored dot
              SPRITE(4, W1, K)[ b -1 a -.9 z -.001      ]   // shadow 1
              SPRITE(4.5, W1, K)[ b -1 a -.95 z -.002   ]   // shadow 2
       }
}
```



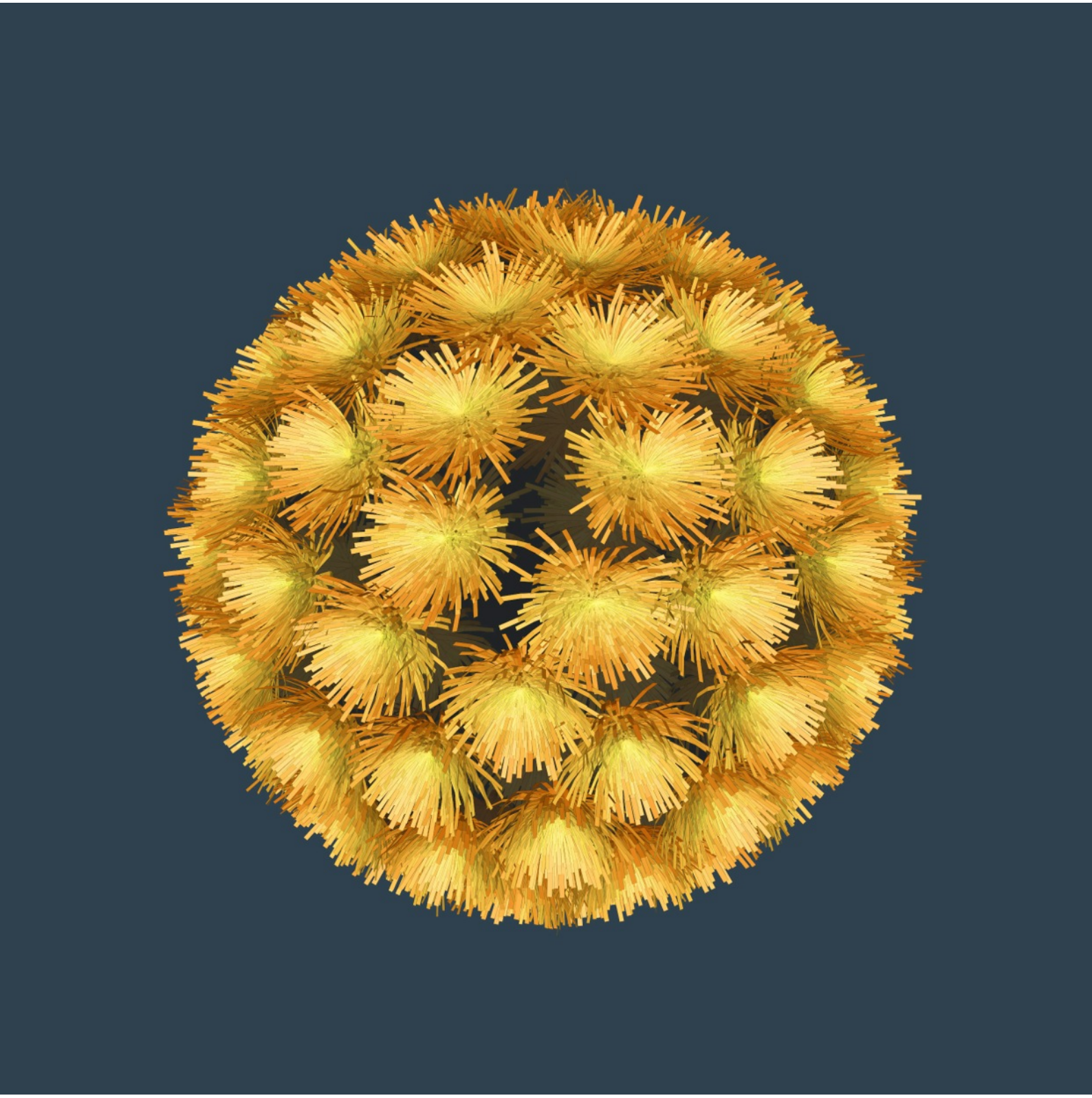
Cubo di cubi

```
shade = 1
import Mycelium3D_2014.cfdg

CF::Background = [b 1]

K=Camera(25, 66, 90)

startshape Matrix(10, Sys())[h 120]

shape Matrix( NN  , vector18 W)
{
	loop i=NN [] loop j=NN [] loop k=NN [] {
		W = MoveXYZ(i, j, k , W)
		if (rand(NN*2)>(i+j))
			CUBE(.97, W, K )[b 1 ]
		else
			CUBE(.88, W, K )[b rand(1) sat 1 a -.5 h rand(60) ]
	}
}
```



Tubi a spirale

```
shade=.95
import Mycelium3D_2014.cfdg

CF::Background = [b 1]
K=Camera( -25, 0, 50)

startshape Two(Sys())

shape Two (vector18 W) {
	RandomPath(W)[]
	W = RotZ(180, W)
	W = ScaleZ(1, W)
	RandomPath(W)[]
}

shape RandomPath (vector18 W) {

	W1 = RotZ(8, W)
	W1 = RotY( .5 , W1)
	W1 = MoveY(.5, W1)
	W1 = Scale(.985, W1)

	if (fuzzy(.015))	//	fuzzy(N) = (rand() < N)
	{
		W1=ScaleX(-1,W1)
		RandomPath (W1)[s MinSize(.02, W) h .5]
	} else
		RandomPath (W1)[s MinSize(.02, W) h .5]

	TUBE(.75, W, W1, K)[ b 1 a -.0 sat 1 ]
}
```



Dente di leone palla

(immagine pagina precedente)

```
shade = .7
import Mycelium3D_2014.cfdg
K=Camera(0, 90, 1580)

CF::Background = [b -.7 sat .5 h -140 ]
CF::Size = [s 110 ]

startshape Centro (Sys())[b 1]

shape Centro(vector18 W)
{
	SPRITE(65, W, K)[b -.7 a -.15]
	count=80
	loop i=1,count/1 [] {
		kz=asin(-1 + 2*i/count)
		W1=RotX( (phi*i) , W)
		W1=RotZ( (kz) , W1)
		W1 = MoveY(35 , W1)
		W1=Scale(.15, W1)
		Flo (400,   W1)[]
	}
}

shape Flo (Q, vector18 W)
{
	W=Stiff(.95,W)
	loop i=Q [] {
		W=RotY(i*phi, W)
		W=RotZ(-90, W)
		W=Scale(i/Q+.1, W)
		W=MoveX((i)/30, W)
		Hair(W)[h -(20*i/Q)]
	}
}

shape Hair(vector18 W)
{
	W1=W
	W=RotZ(rand(0,25), W)
	W=MoveY(16, W)
	TAPE(1.5, W, W1, K )[ a 1 b 1 sat 1 h 60]
	TAPE(1.7, W, W1, K )[ a -.5 b -1 sat 1 h 60 z-.1]
	Hair(W)[s MaxAge(6, W)]
}
```

3D in context free art

Context Free Art è un programma di disegno 2D, questo significa che non potremo effettuare le tipiche operazioni da programma di modellazione 3D come ruotare una vista in tempo reale oppure scegliere un algoritmo di rendering.

3D con context free significa avere a disposizione una geometria 3D che sarà rappresentata con “l’algoritmo del pittore”. Luci e ombreggiature sono perciò anch’esse da “disegnare”.

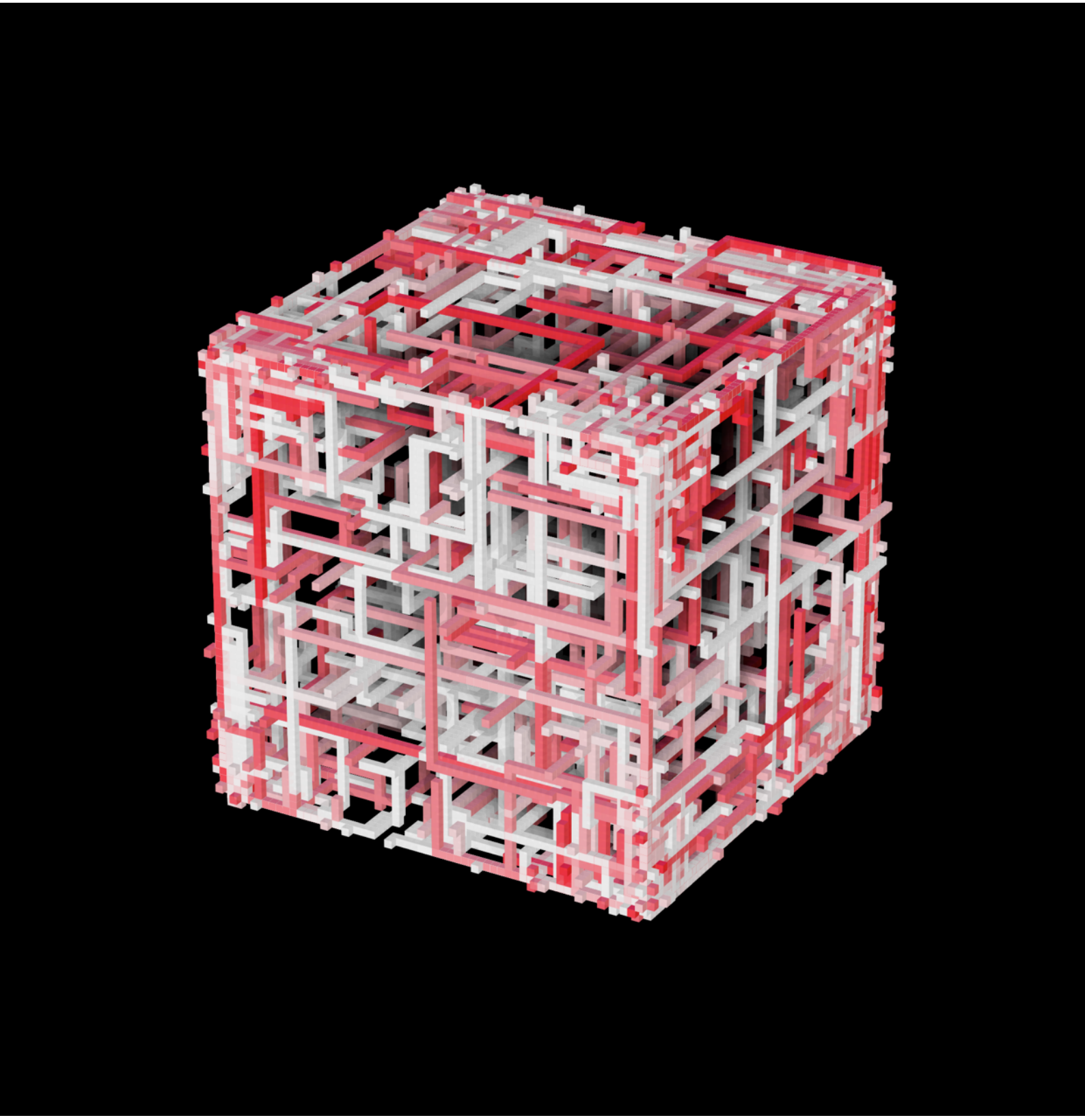
Questi strumenti che appaiono molto limitanti per usi di grafica 3D tradizionale possono essere invece potenti e unici per applicazioni dove è richiesta computazione di un numero elevatissimo di elementi, notevole velocità di disegno, o uso massivo di trasparenze.

Applicazioni possibili sono ad esempio: grafica generativa, frattali, simulazioni di strutture naturali, animazioni di strutture complesse.

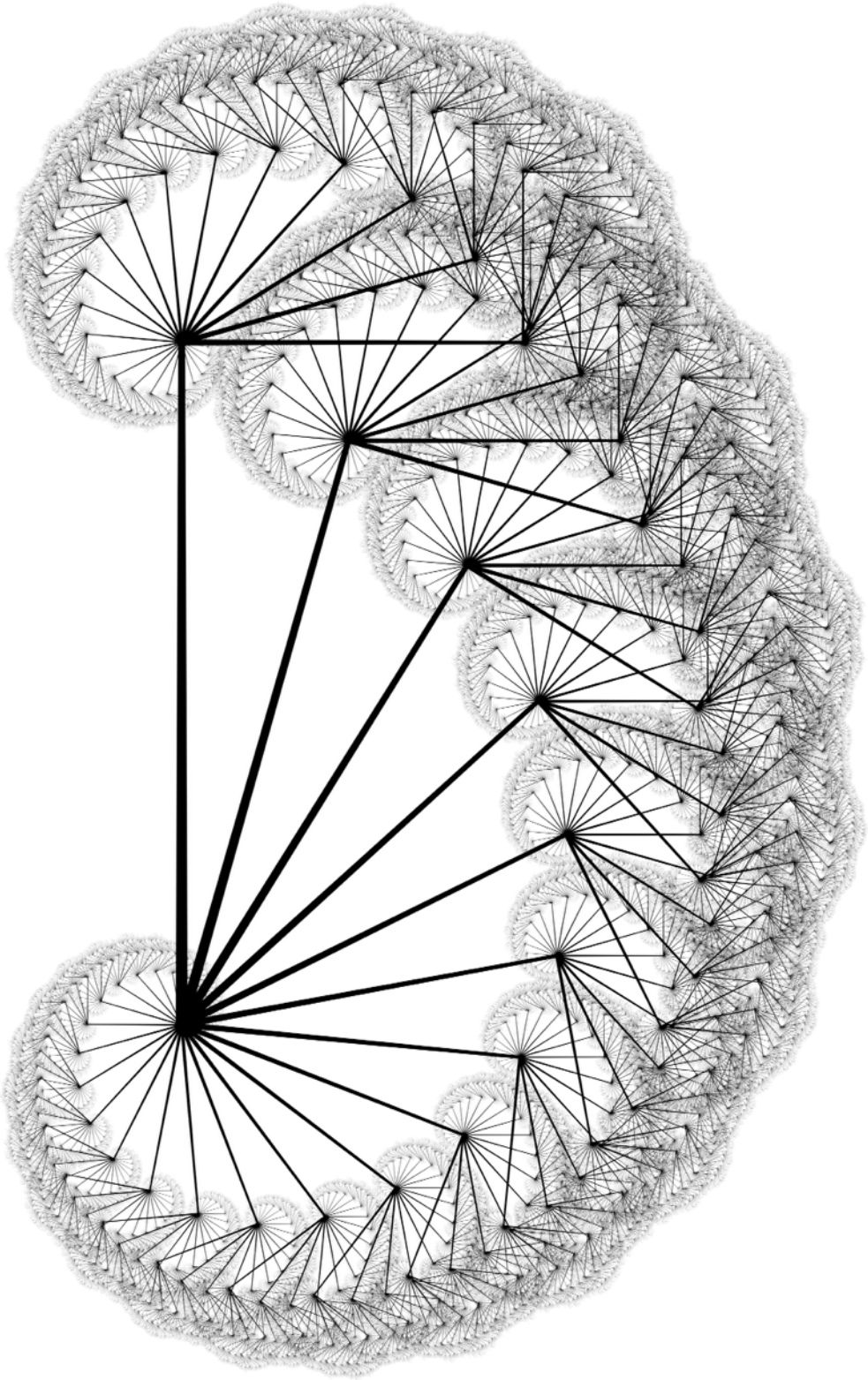
Algoritmo del pittore e occlusione ambientale

L’algoritmo del pittore mette in fila gli elementi da disegnare, dal più lontano al più vicino e diligentemente li rappresenta sovrapponendoli.
Non è particolarmente ottimizzato ma ha la caratteristica unica di potere gestire infiniti livelli di trasparenza. Il canale di opacità a 16 bit di CFA ne esalta questa caratteristica: permette, ad esempio, di realizzare effetti di ombre molto espressivi e spesso simili a quanto definito altrove come “occlusione ambientale”.

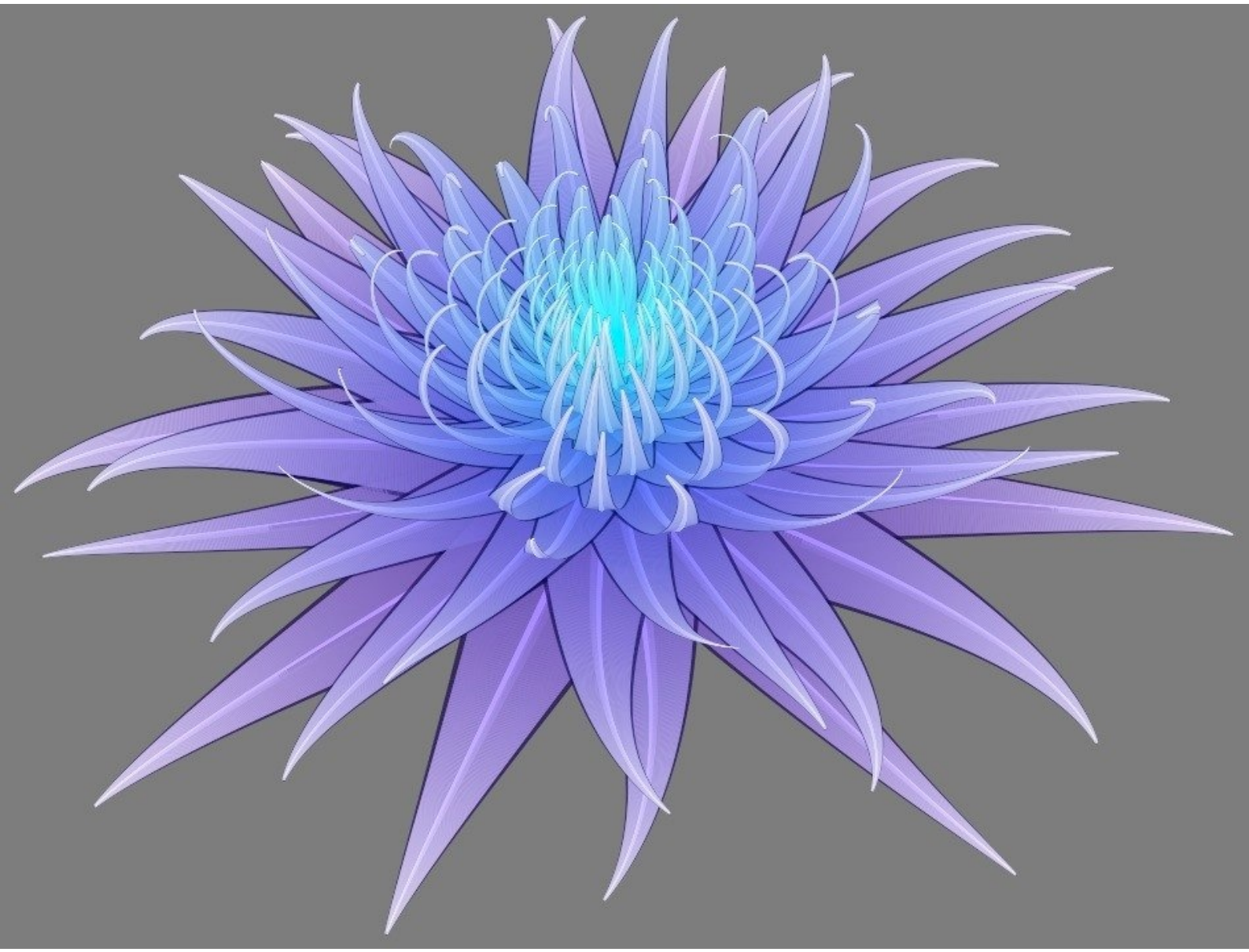
Immagini realizzate con Mycelium 3D



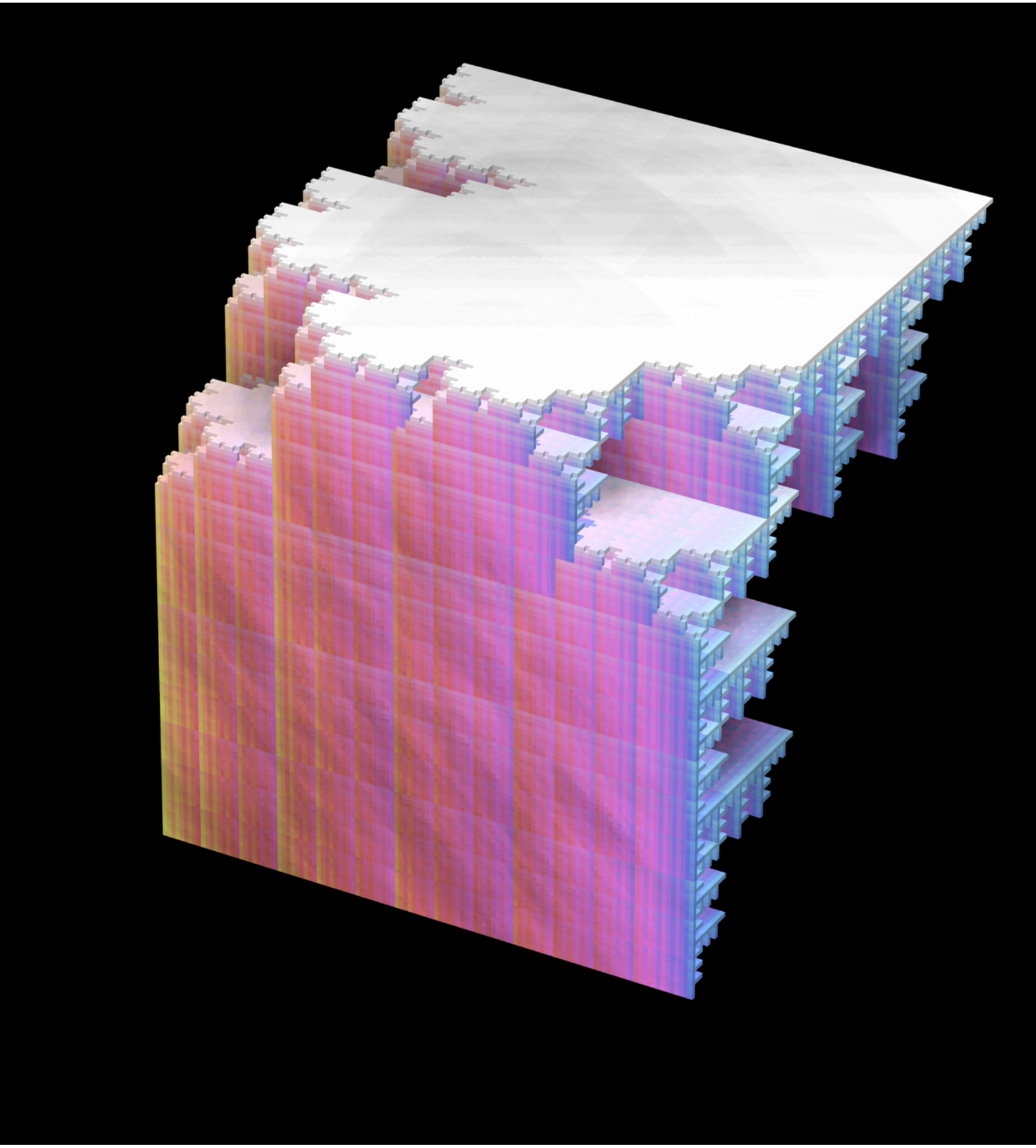
L’effetto di occlusione ambientale è ottenuto sovrapponendo Sprites semitrasparenti.



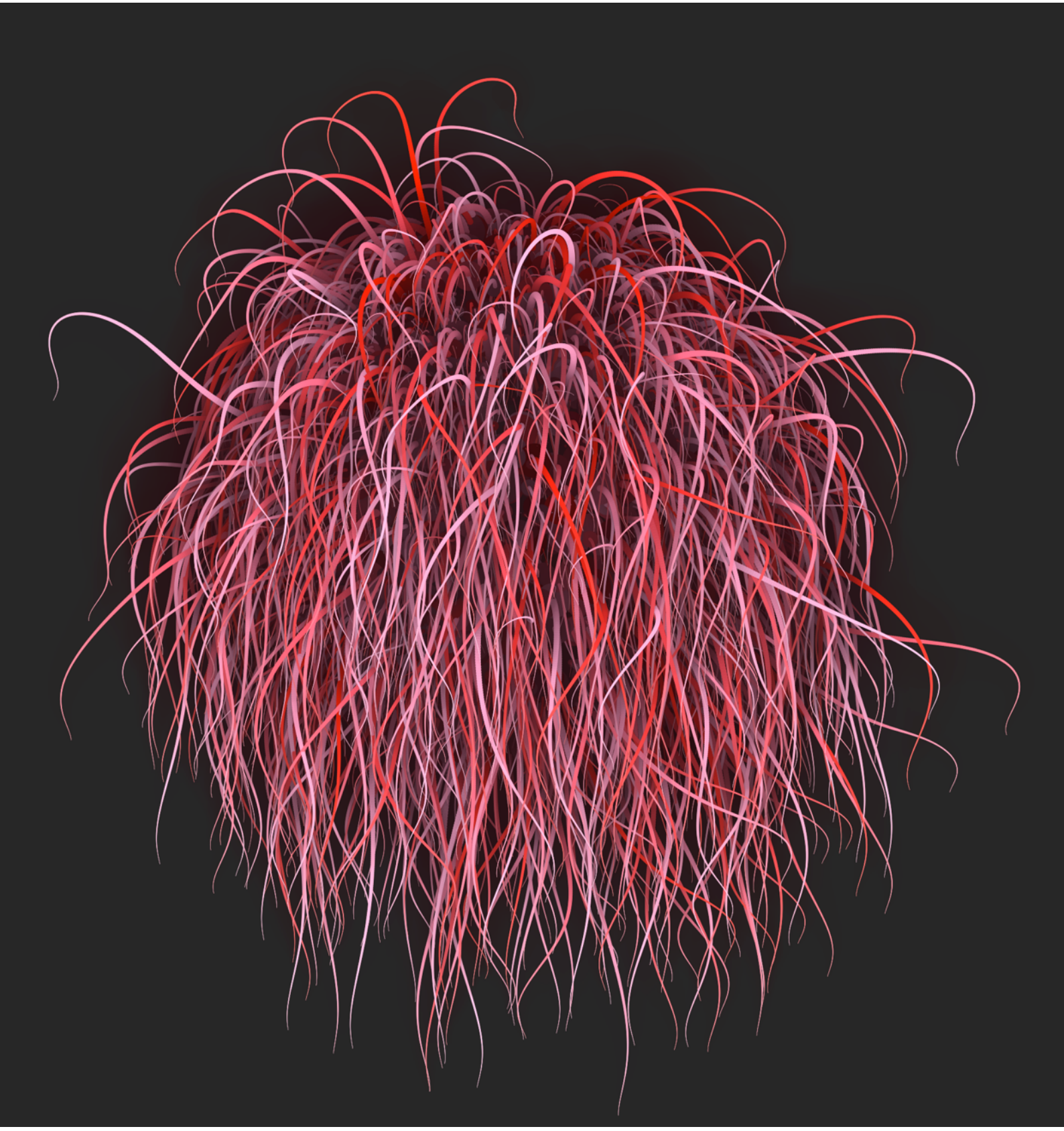
Uso delle linee per tracciare un frattale 2D.



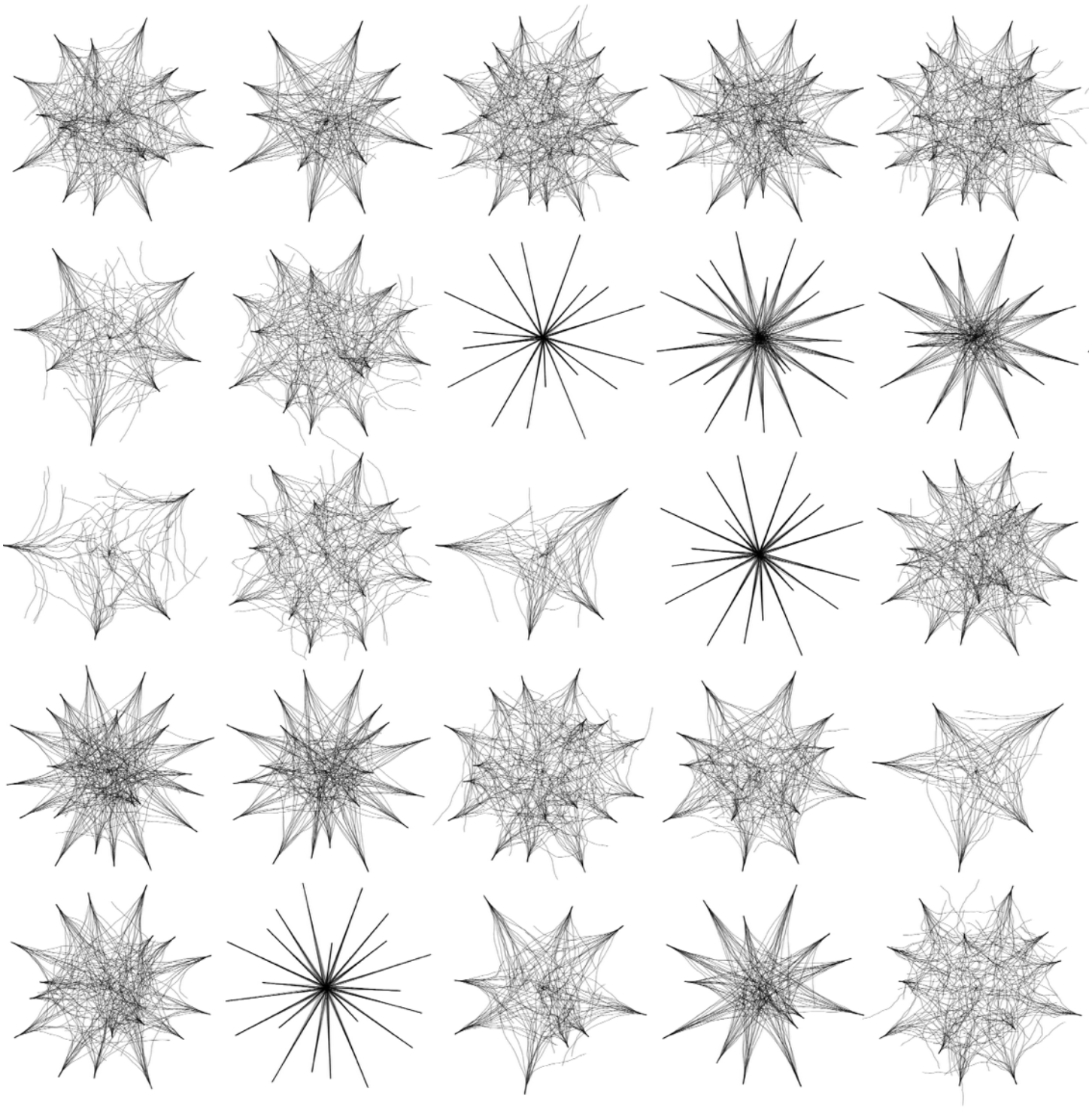
Struttura naturale



Frattale 3D (cubo di Menger generalizzato)

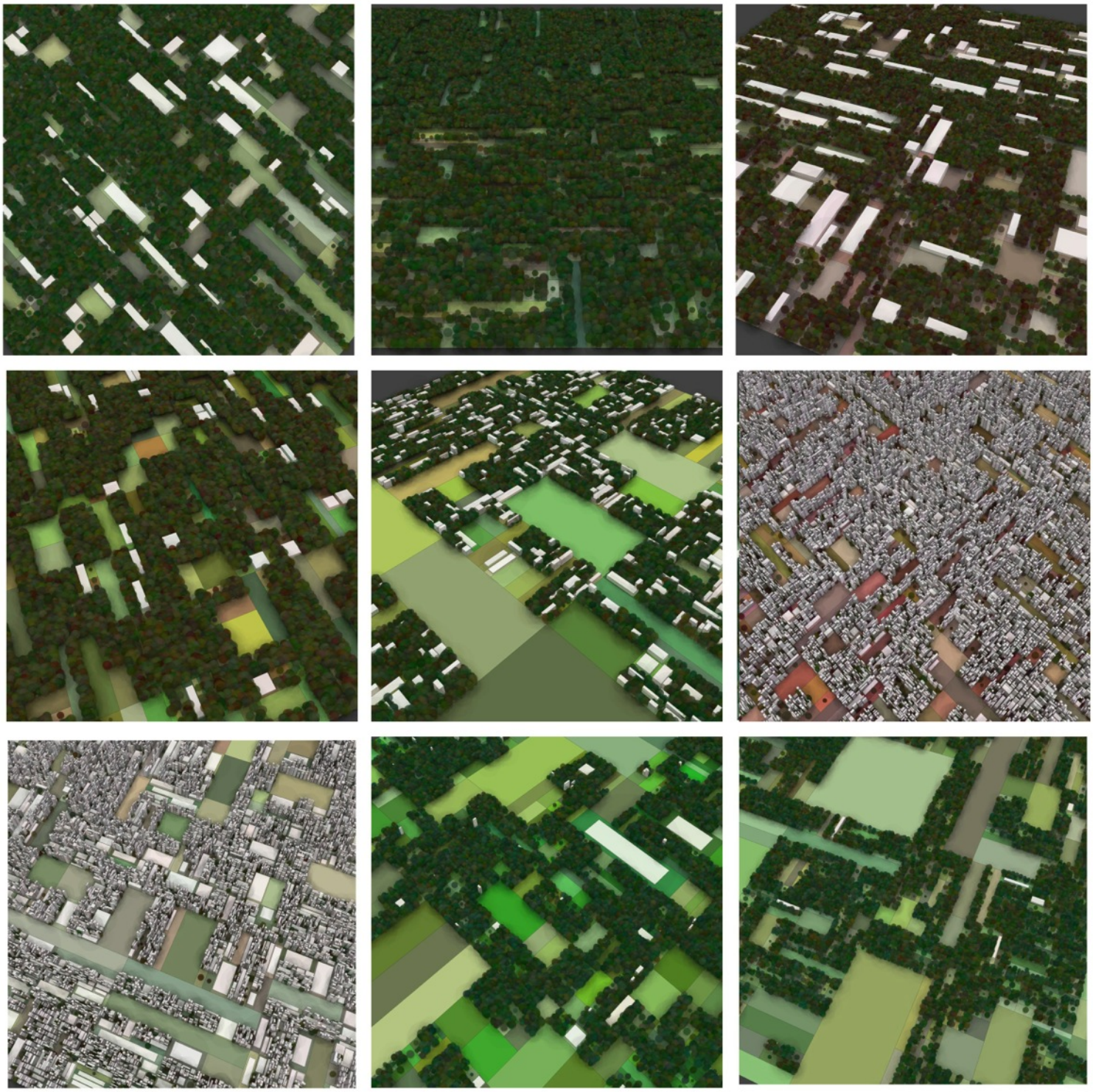


10.000 hairs / 1.500.000 shapes



Stelle tridimensionali basate su angolo aureo phi.

Simulazioni naturali



Paesaggi generati con un unico algoritmo iterativo.

Licenza creative common

Licenza standard

Mycelium3D.cfdg di arch. Federico Donelli è distribuito con Licenza Creative Commons Attribuzione - Condividi allo stesso modo 4.0 Internazionale.
Per leggere una copia della licenza visita il sito web http://creativecommons.org/licenses/by-sa/4.0/

Licenze diverse

Qualora fosse necessaria una differente licenza d'uso è possibile farne richiesta all'autore (arch. Federico Donelli). Contatti presenti sul sito www.fdsa.it