Computer Graphics
Generative Models

Dr Simon Lock
simon.lock@bristol.ac.uk

FYI - Processing

- Topics will be introduced using various animations
- Created using a tool called “Processing”
- Very popular with digital artists

- Java-based programming environment
- With it’s own built-in hardware accelerated rasteriser !
  (So we don’t have to use the one I built in C ;o)

- I’ll use it as a sandpit for demonstrating key concepts
- Experimenting with interactive graphics…
- Displaying things and then “prodding” them

Purpose of Session

- Working with pre-built models is all well and good
- Problem is they can be time-consuming to create (3D scanning, Hand-crafting with applications etc)

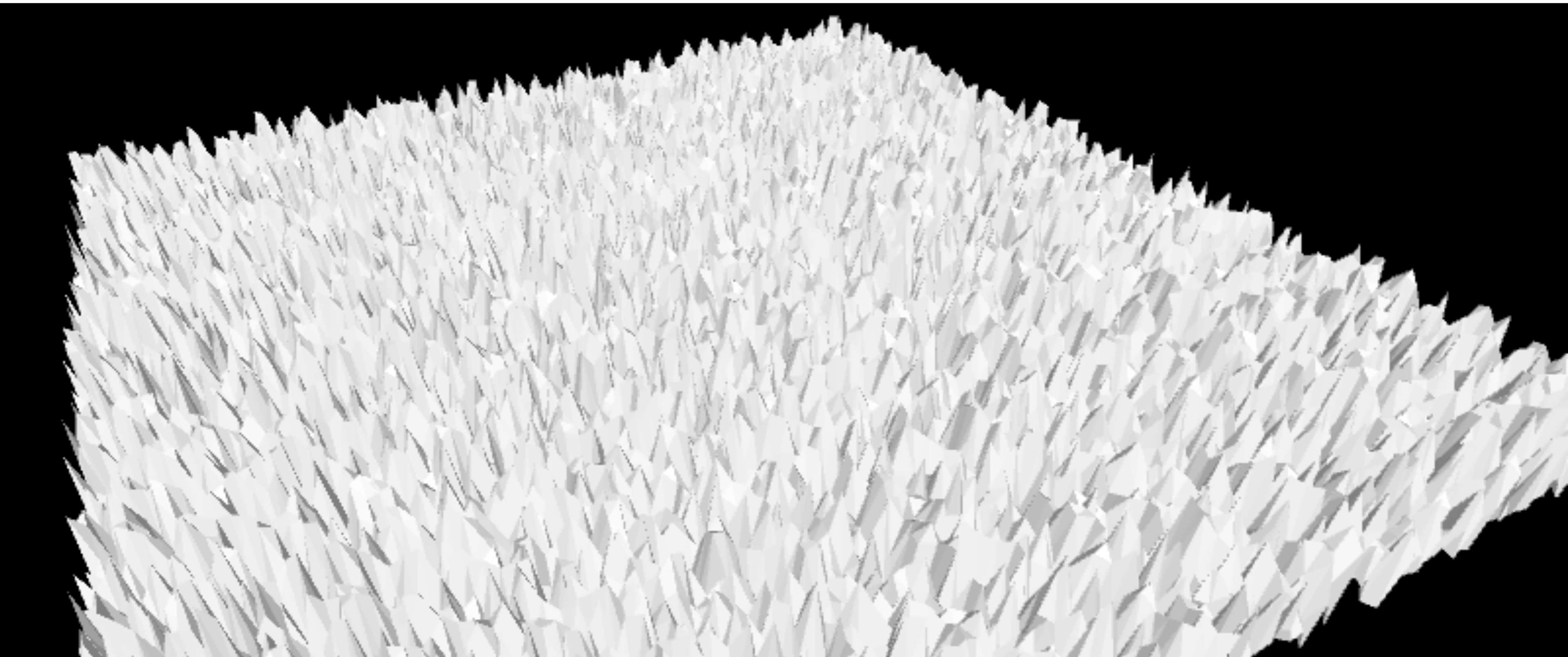
- Additionally, they are by their very nature ***finite***
- What if we wanted ***unlimited*** artefacts ?
- The answer is to write code to generate models !

- Might make a nice extension for your project !!!

Generative Landscapes

First Attempt

- A naive approach might be to randomly the set height of each point of land
- As we can see, this doesn't work particularly well…

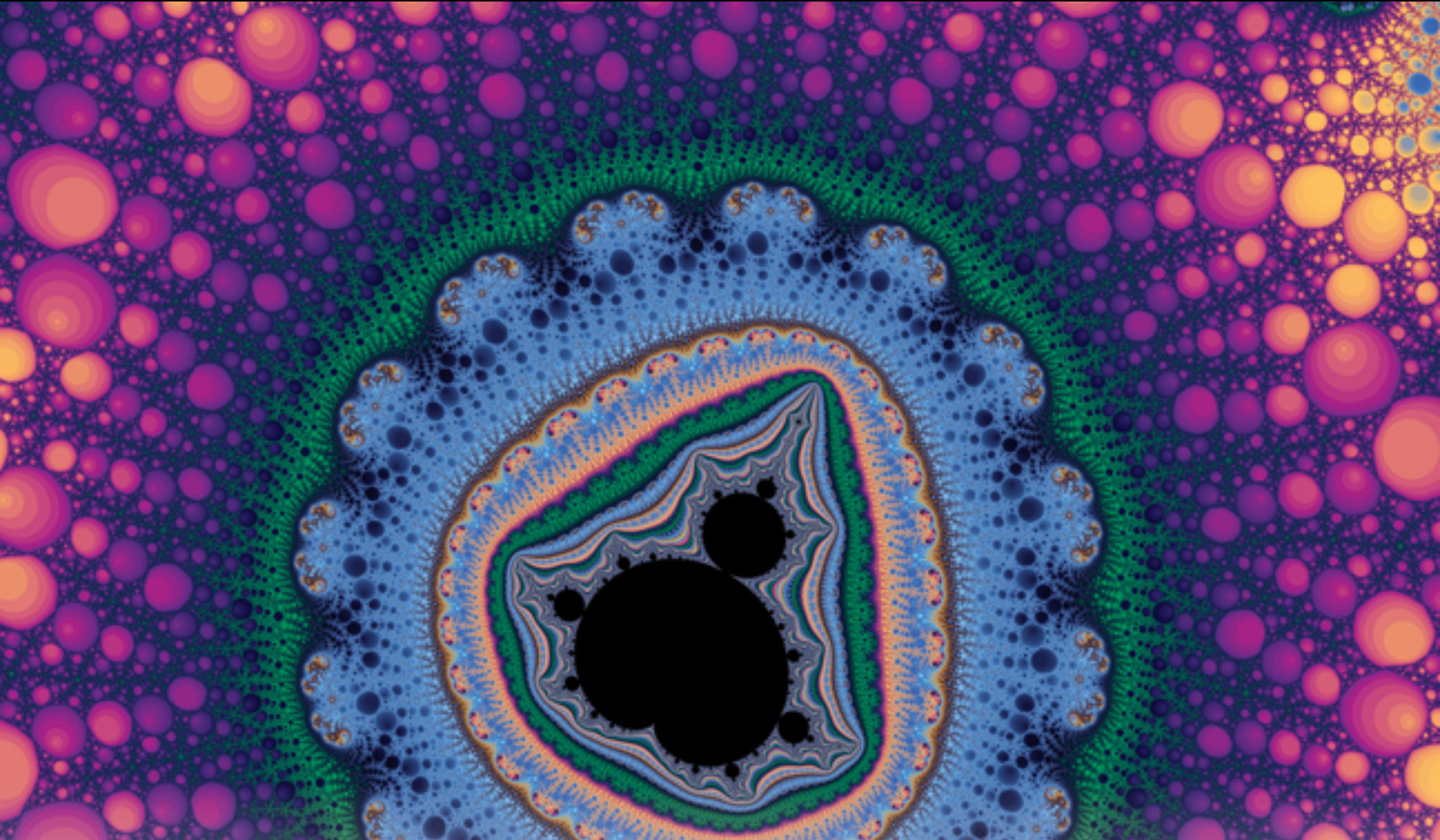


Revised Approach

- What we need is some kind of “smoothing”
- Something that will remove the extreme spikes
- But that will leave “undulations”

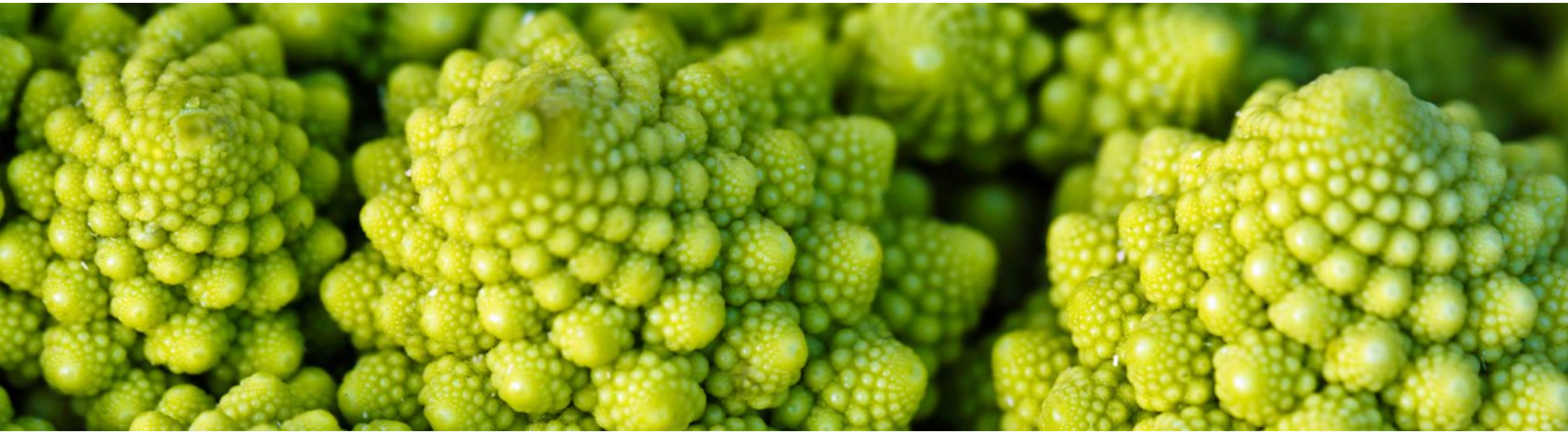
- Something that will look like real land !!!

“Fractal” Landscapes



In what way are they fractal ?

- Structure defined by a formula/algorithm
- Potentially infinite level of detail
- Created with an iterative or recursive algorithm
- Parallels with / similarity to nature


Romanesco Broccoli

Basic Concept

- Basic concept is to first randomise a grid of points
- Then incrementally average out the heights
- Starting out with a big “stride”
- Then getting increasingly finer grained

- It’s a lot easier to explain with an animation !
- Let’s see how it works in 2D first (for simplicity)

[Terraform2D]

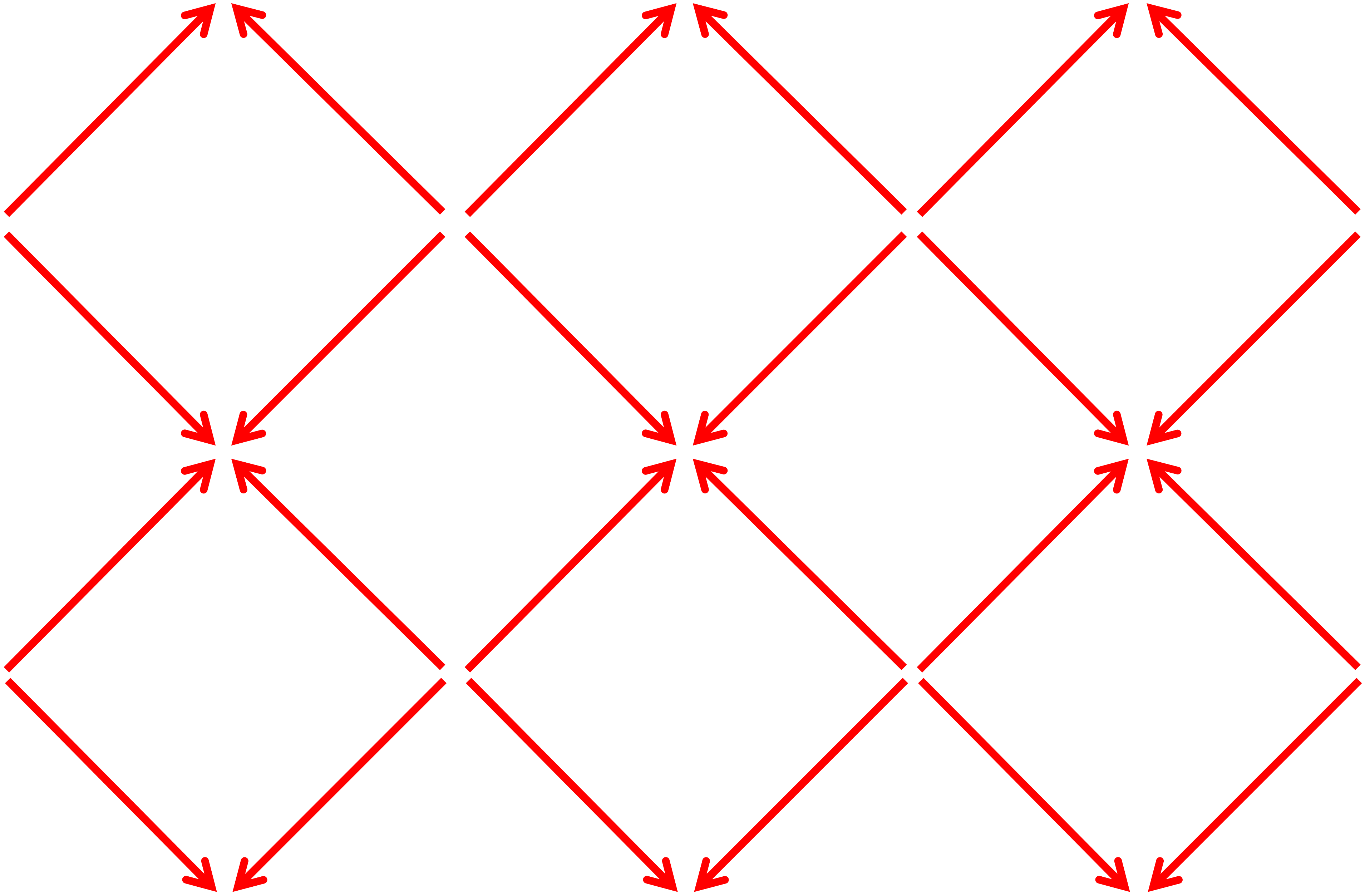
Diamond-Square Algorithm

- Commonly used fractal landscape generator
- Good sophistication / understandability balance

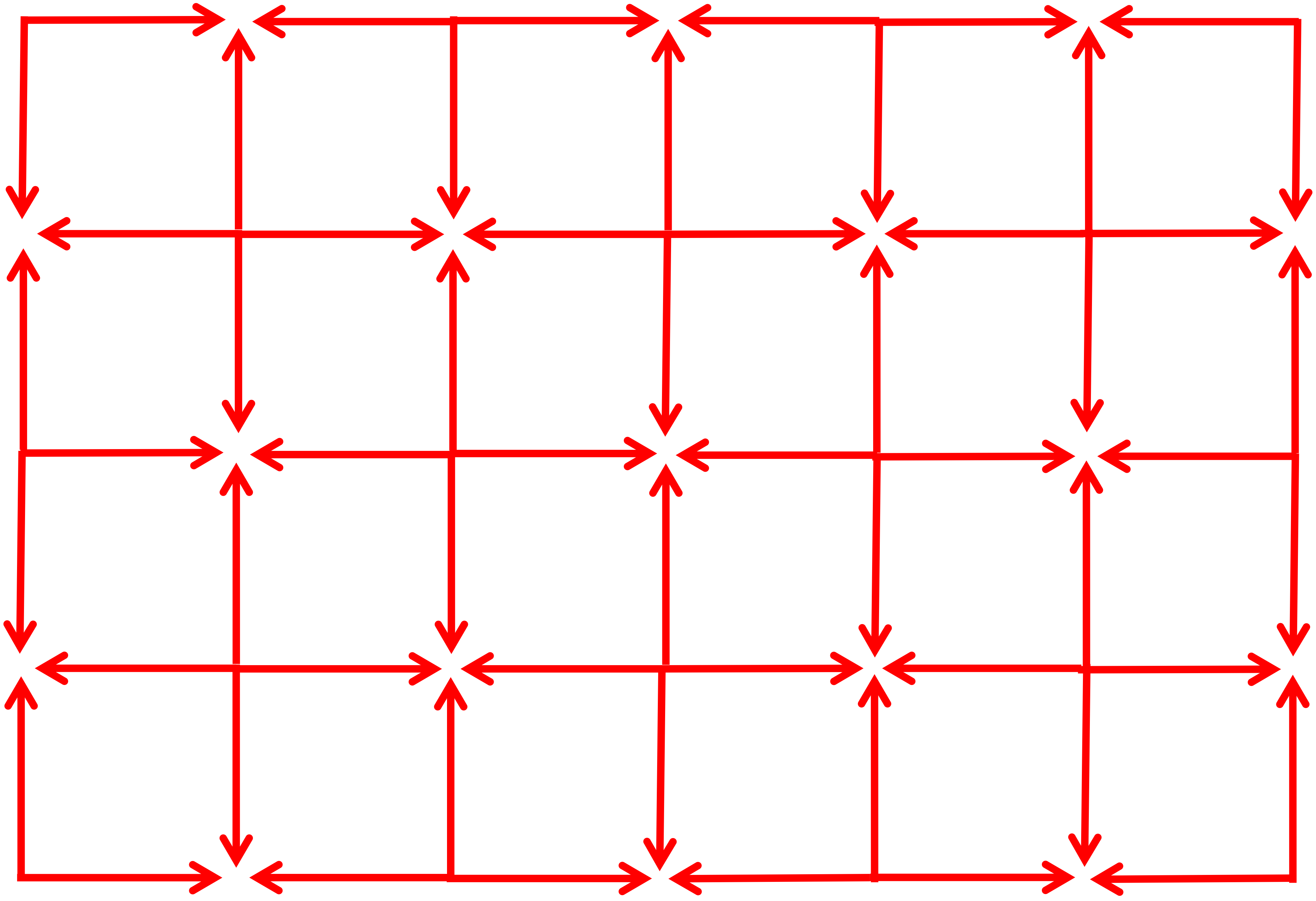
- Variants do exist: everyone has their own tweaks
- We will explore a fairly basic version
- But one that produces some nice-looking results

- Works like 2D version, but with an extra dimension !
- Needs to average out values in multiple directions
- Involves two steps: Diamond Step & Square Step…

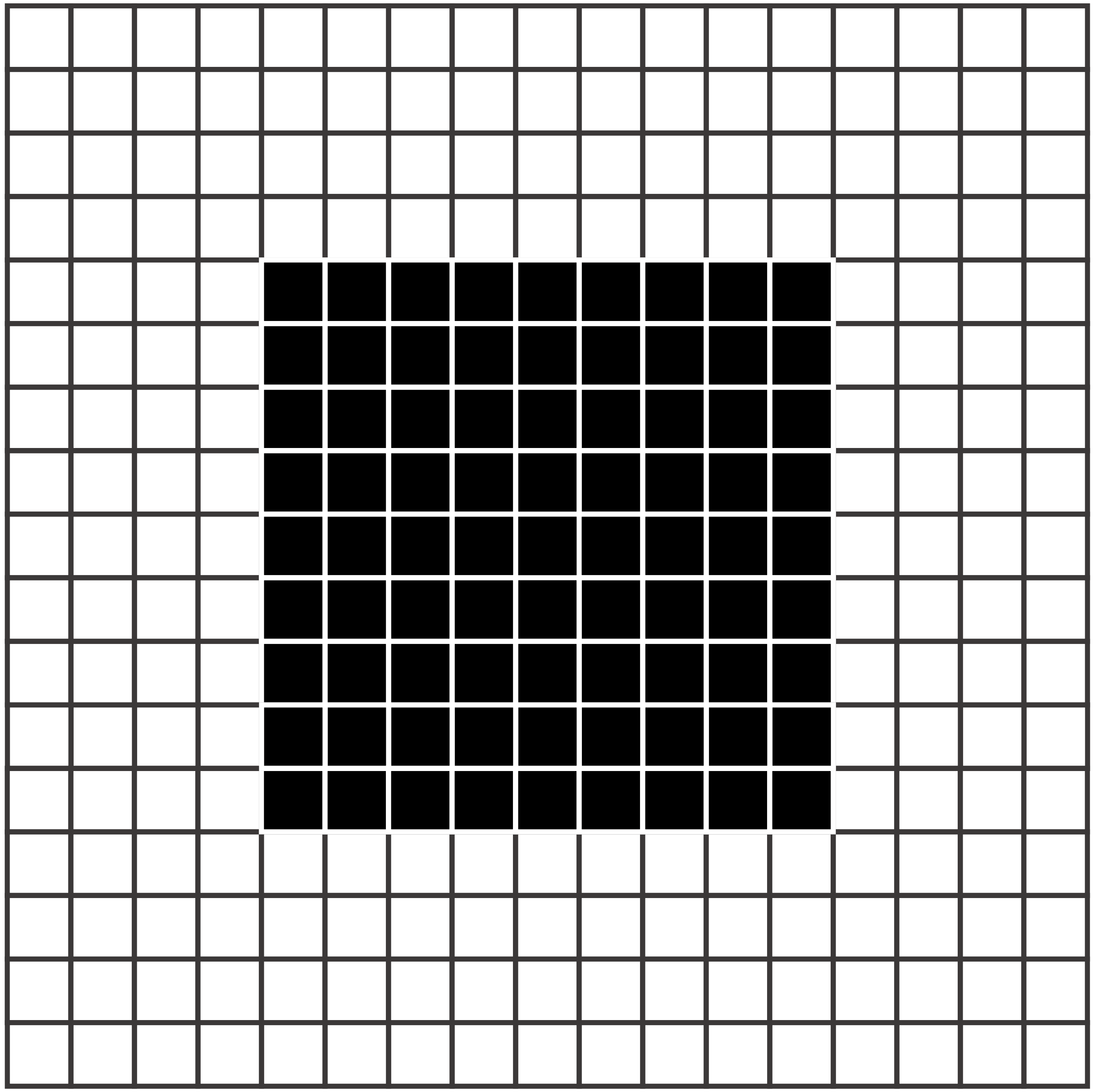
Diamond Step Averaging Pattern



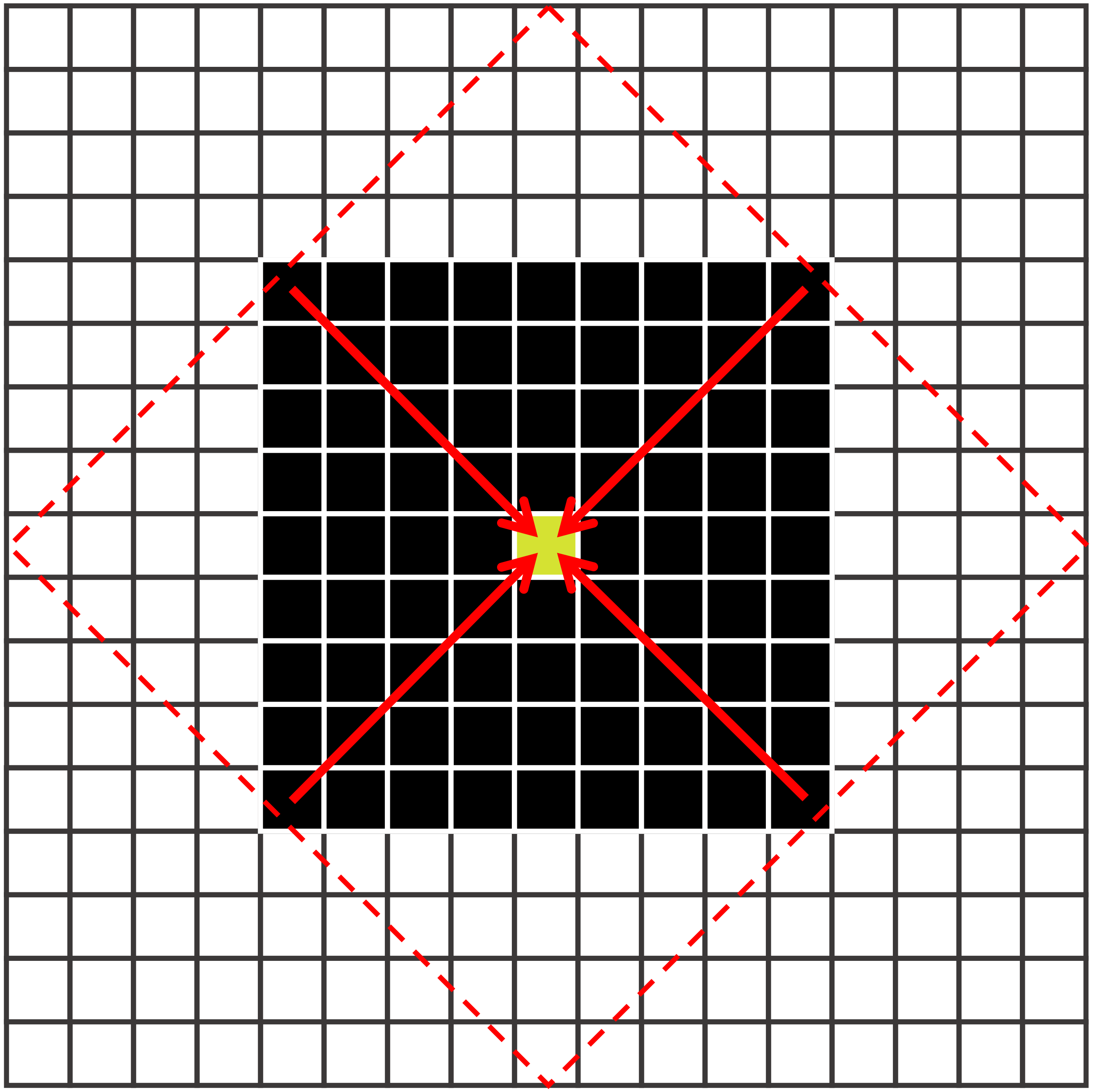
Square Step Averaging Pattern



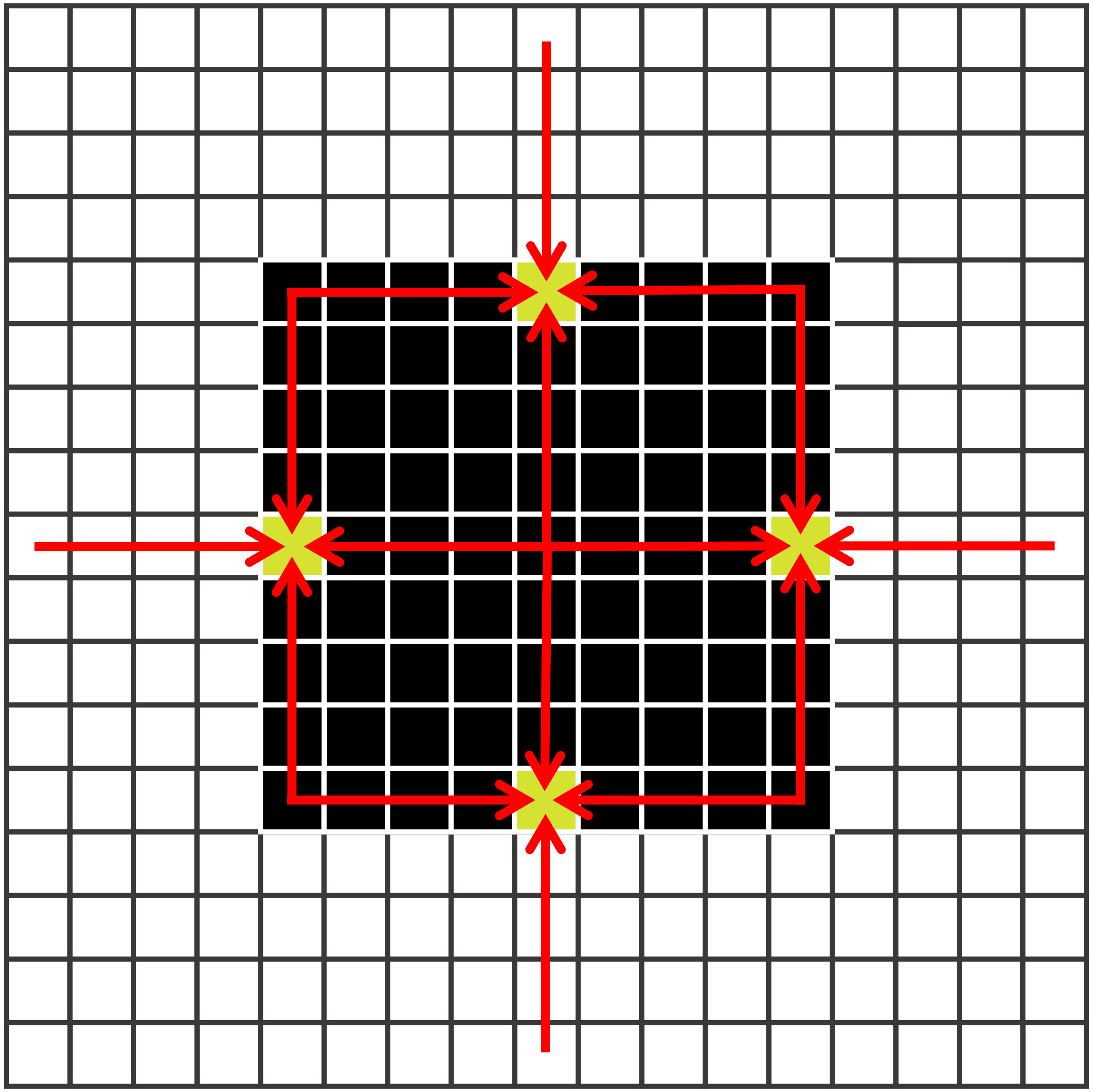
Landscape + Margin



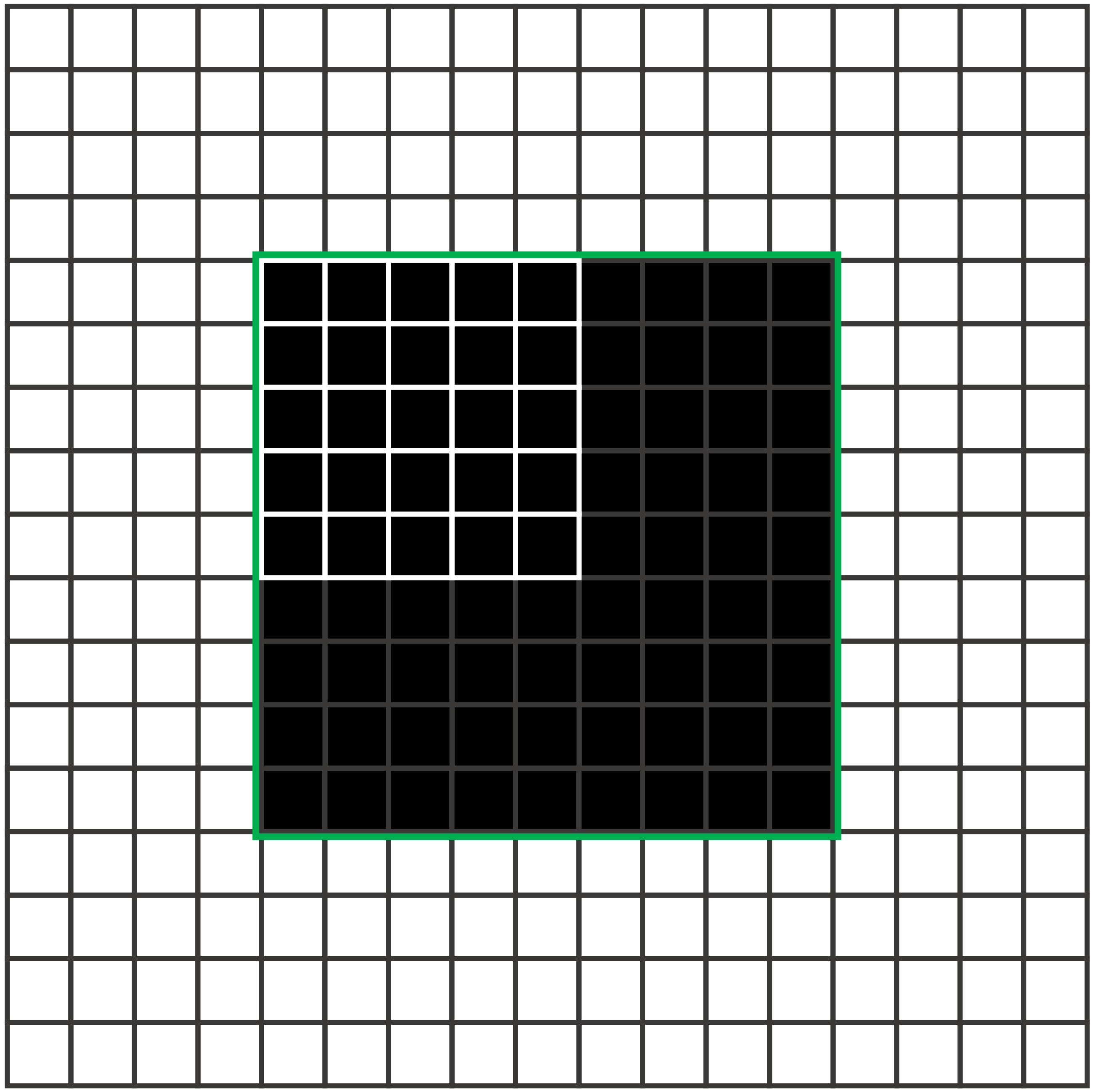
Diamond Step



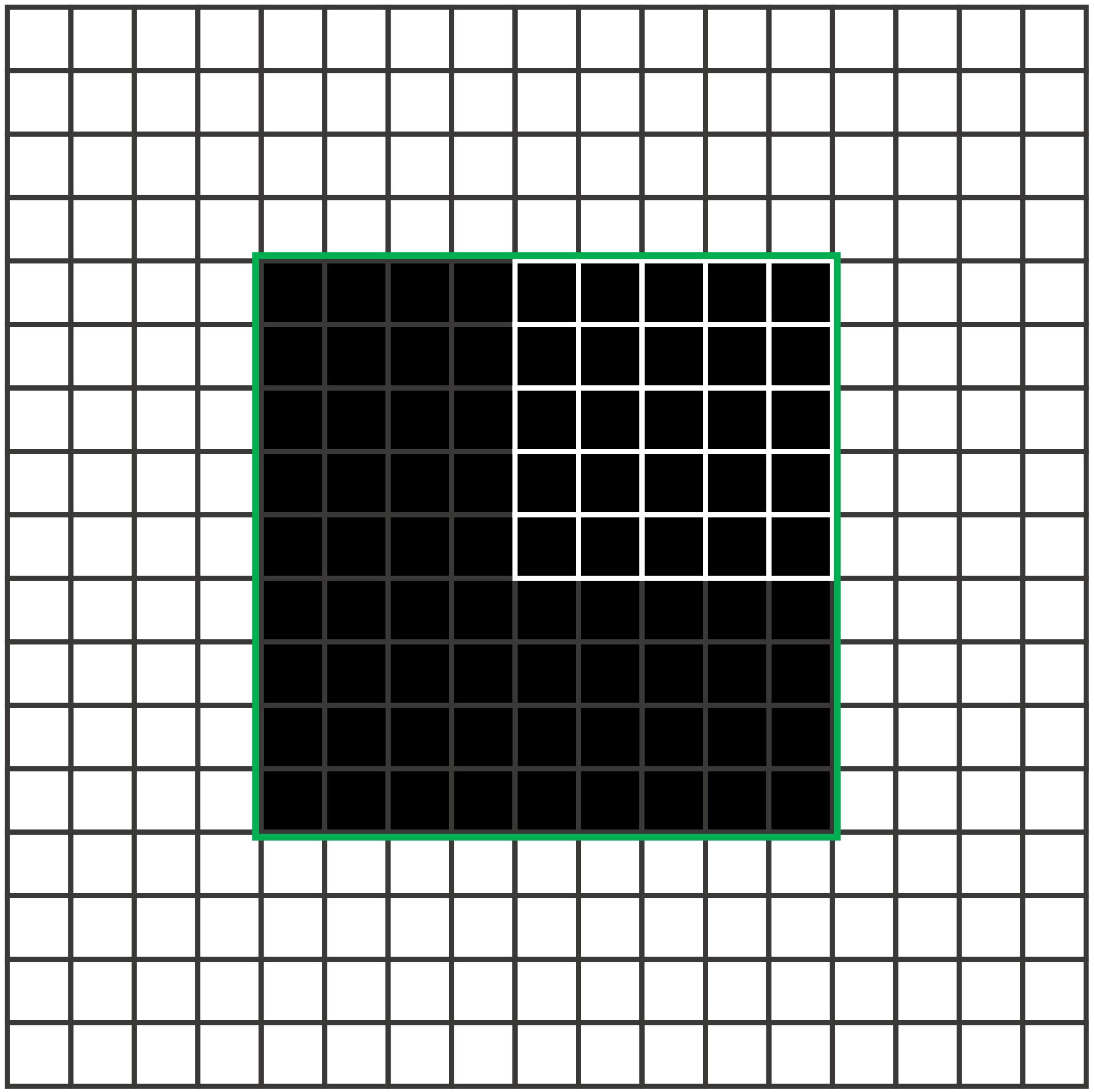
Square Step



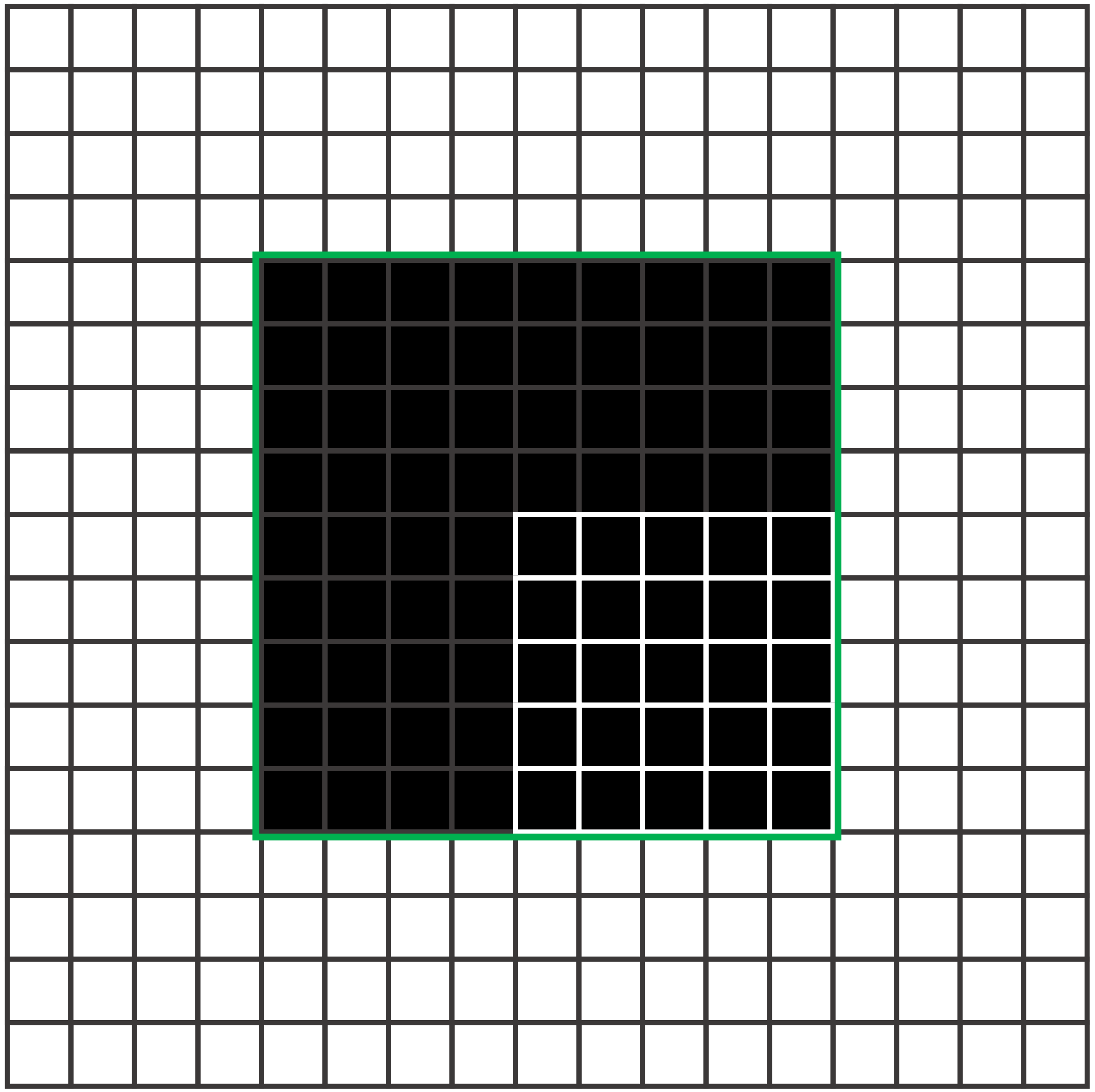
Split



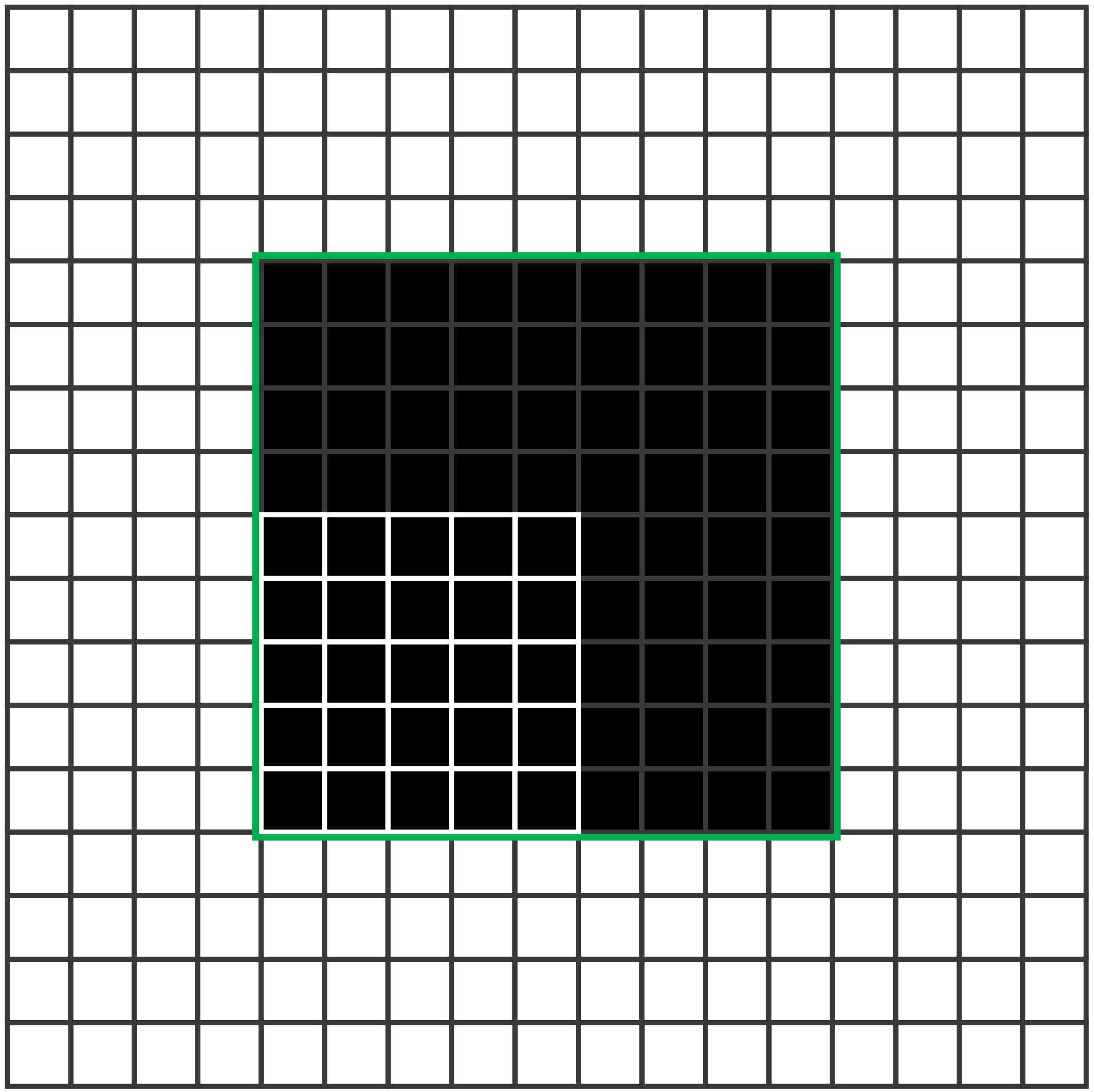
Split



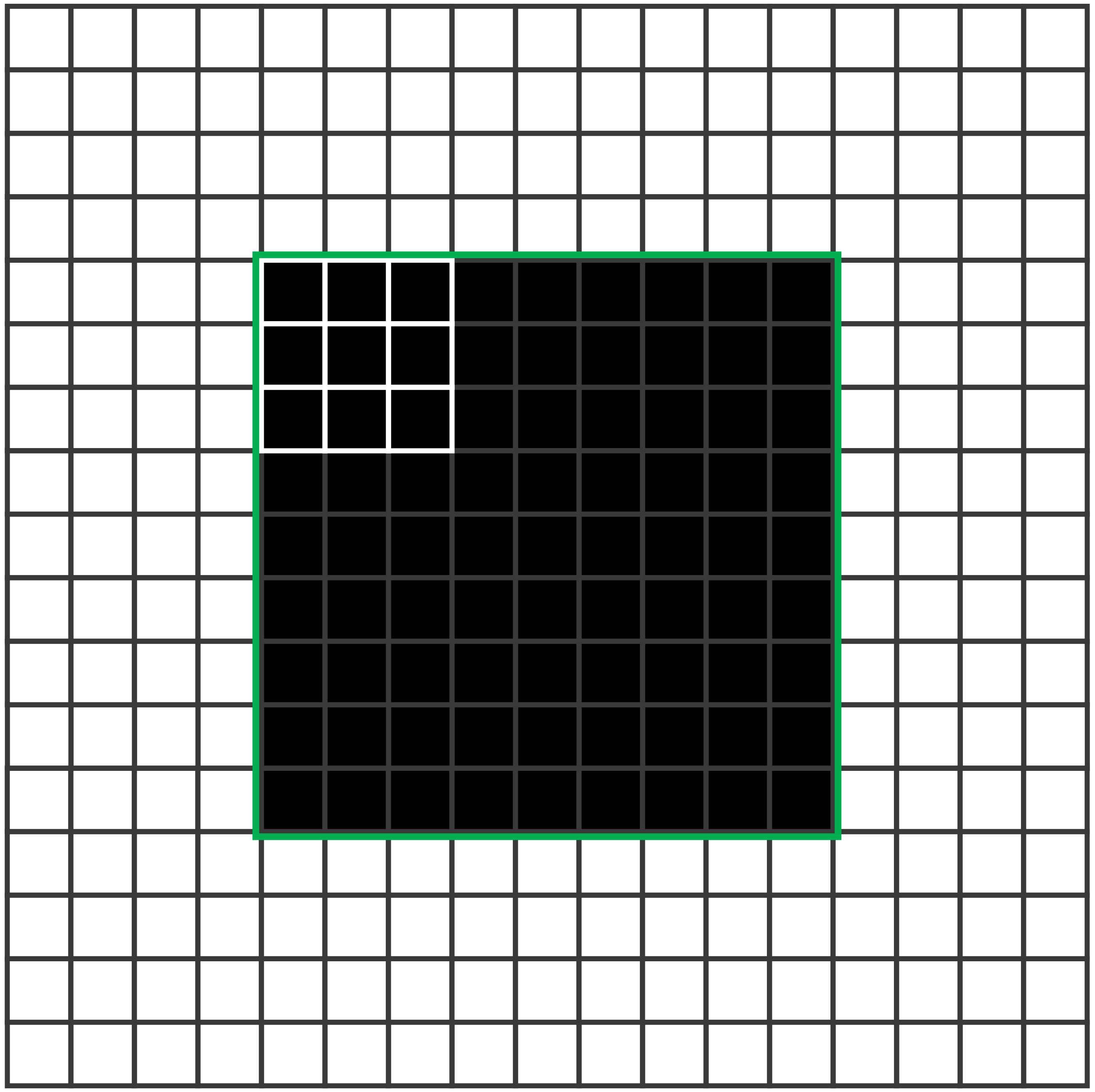
Split



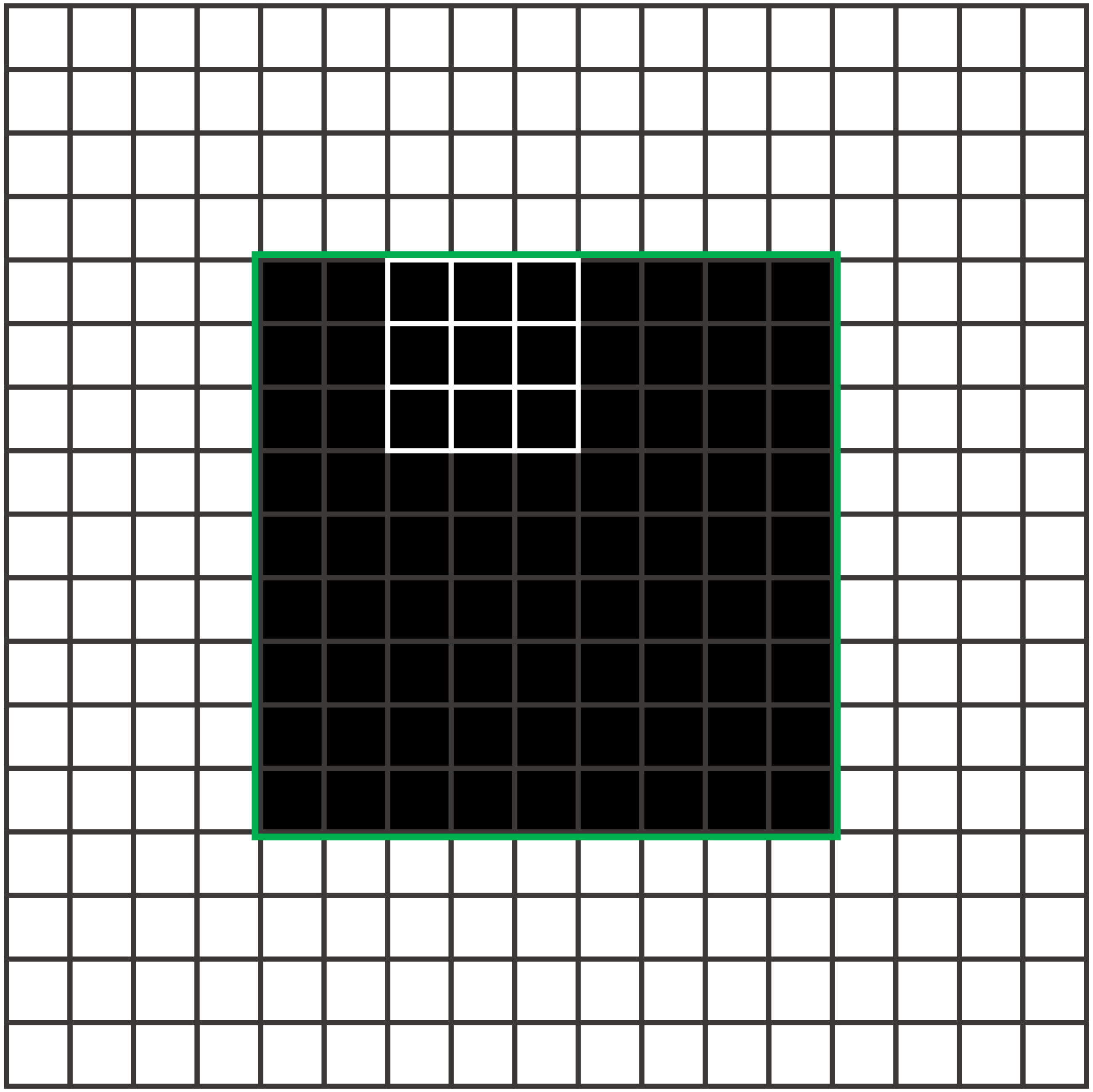
Split



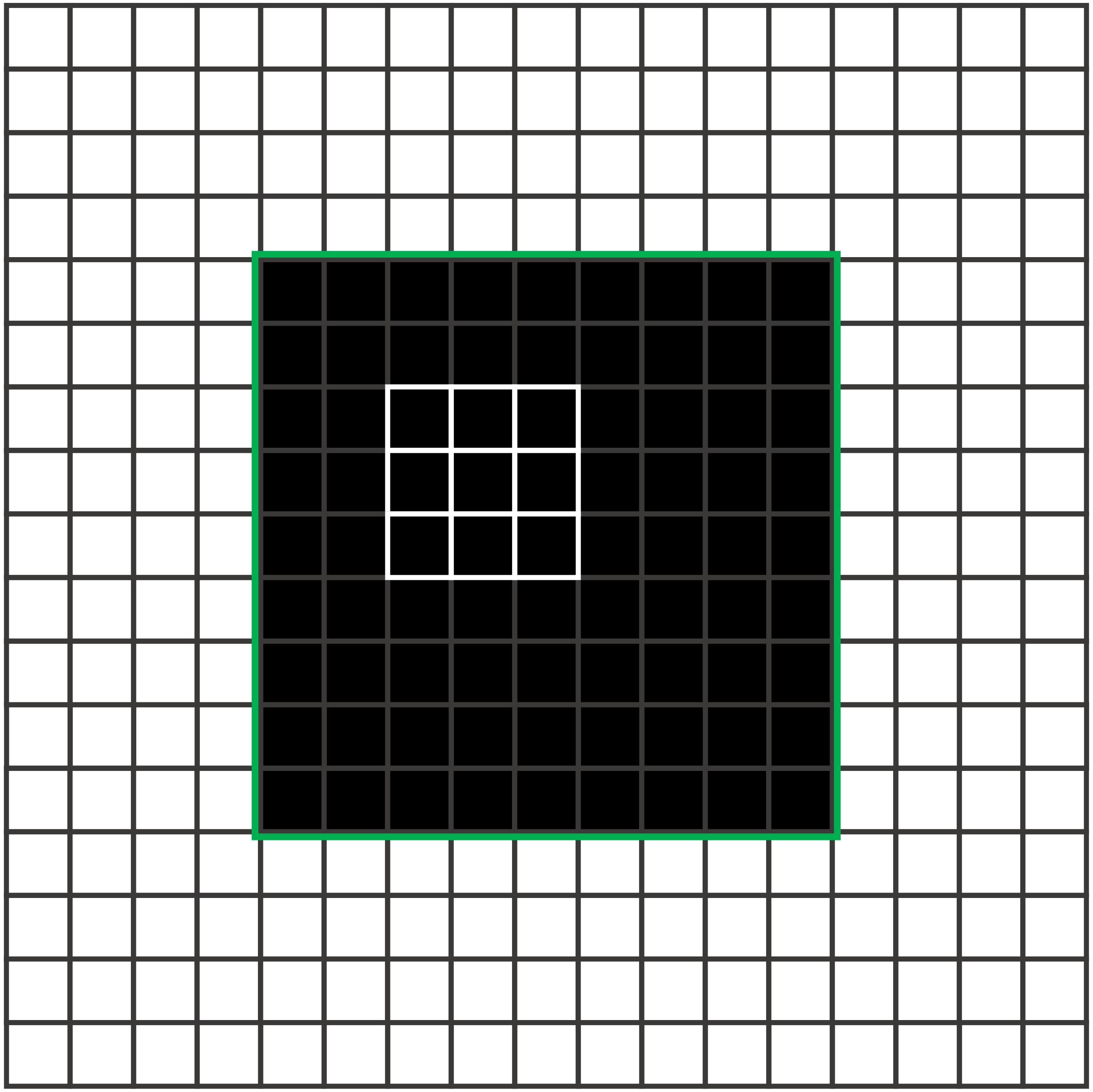
Split Again



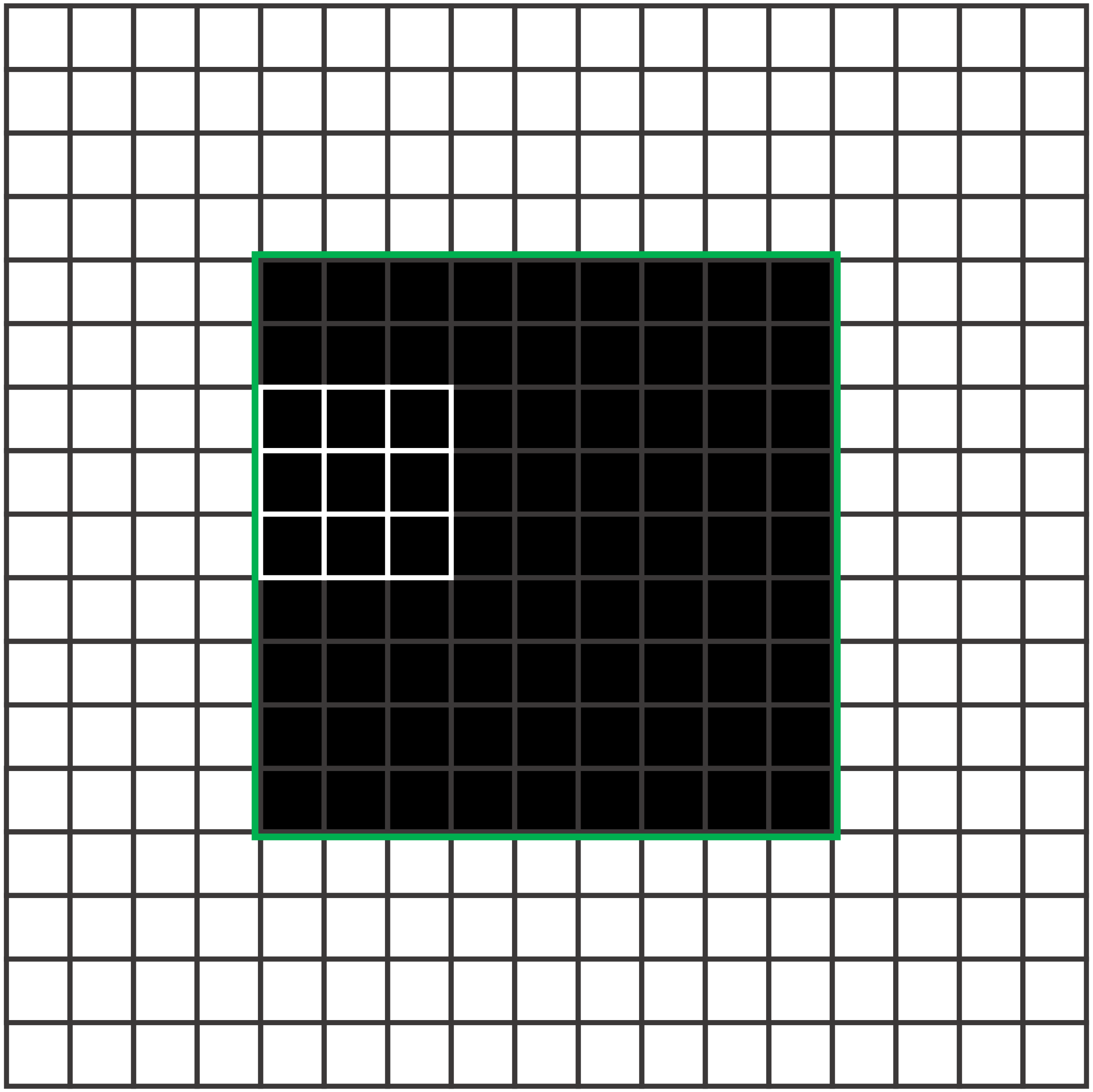
Split Again



Split Again



Split Again



[DiamondSquare]

Hints and Tips

- Use landscape of size $2^n + 1$ (for cleaner halving !)
- Randomise all points to begin with
- Deal with landscape edges (margin/wrap-around)
- Don’t use recursive solution (just an iterative one) !
- Watch out for seams and creases !
- Experiment with the roughness factor
- Watch out for those minus ones !!!!

Replication

Replication

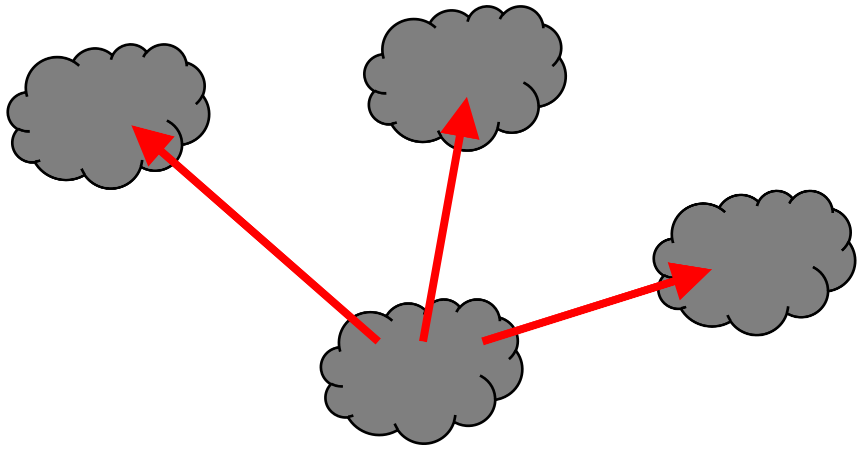
- A key tool for model generation is replication
- It’s very easy to create complex models…
- By just duplicating simple structures

- Easily done with a bit of code…
- Use a loop to create multiple instances
- Use transforms to position them where we want !

Translation

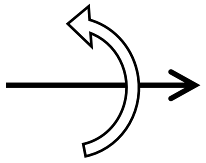
- Translation is the simplest form of transformation
- Involves shifting coordinates in X, Y & Z dimensions

- Rather than moving camera or light (as before)
- Create then shift individual 3D shapes…



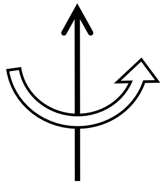
Rotation Matrices

Rotation about X axis



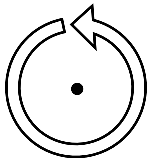
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Rotation about Y axis



$$\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Rotation about Z axis



$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[FlyThrough]

Import/Export of Generated Models

- So we have gone to all the effort creating models
- It would be nice if we could keep these for future

- Load them back in again later on
- Open them up with other applications
- Share them with other people
- Or send them to a 3D printer !

- Various file formats are possible
- OBJ files are versatile and popular…

OBJ File Content

Core content:

  Vertex: Point in 3D space

  Facet / Face: Triangle consisting of 3 vertices)

  Vertex Texture: Reference to a texture map

Other (optional) content is also possible:

  Surface normals

  Material (ambient/diffuse/specular colours)

  Polygon (non-triangular) shapes

OBJ Example

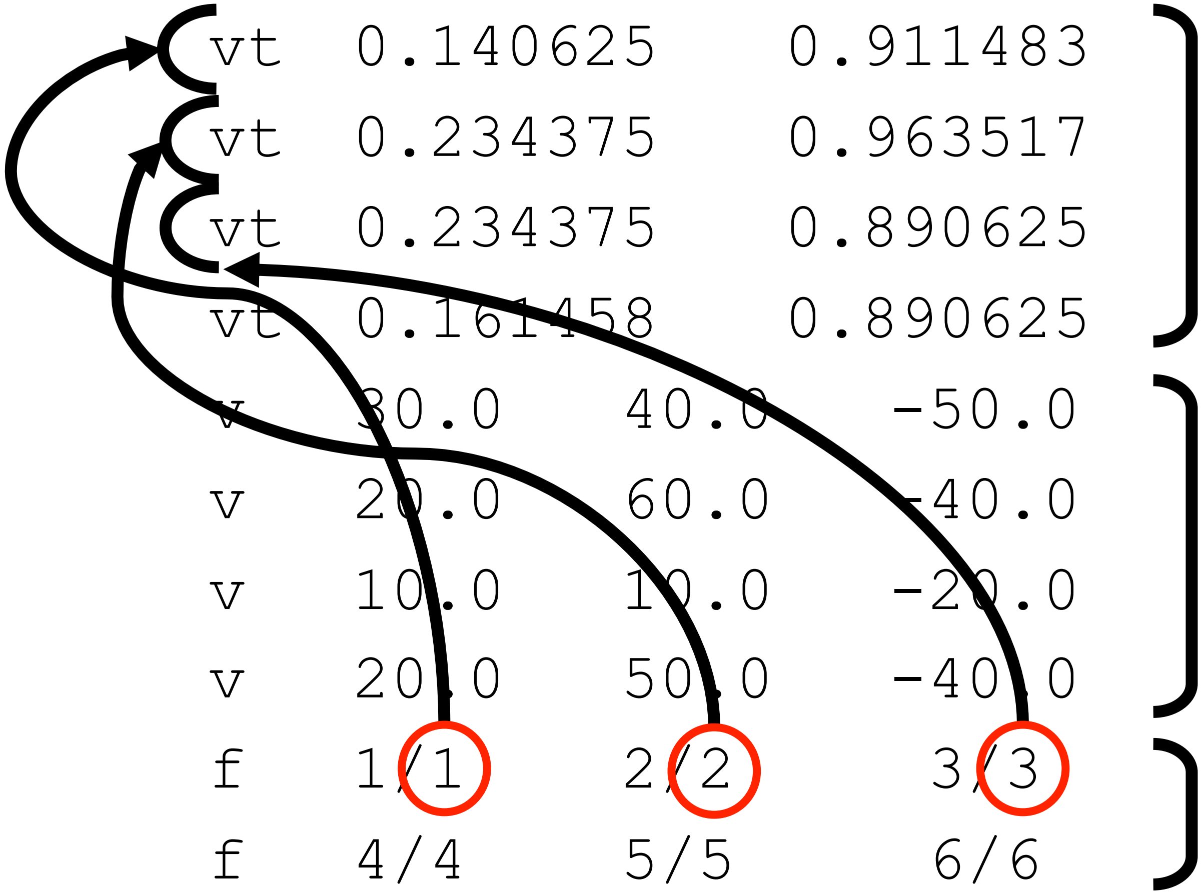
```
vt   0.140625    0.911483
vt   0.234375    0.963517
vt   0.234375    0.890625
vt   0.161458    0.890625
v    30.0     40.0     -50.0
v    20.0     60.0     -40.0
v    10.0     10.0     -20.0
v    20.0     50.0     -40.0
f    1/1      2/2      3/3
f    4/4      5/5      6/6
```

Vertex Textures from image file (see next slide)

3D Vertex Coordinates

Triangular Facets/faces (with indices)

OBJ Example



```
vt   0.140625     0.911483
vt   0.234375     0.963517
vt   0.234375     0.890625
vt   0.161458     0.890625
v    30.0     40.0     -50.0
v    20.0     60.0     -40.0
v    10.0     10.0     -20.0
v    20.0     50.0     -40.0
f    1/1      2/2       3/3
f    4/4      5/5       6/6
```

Vertex Textures from image file (see next slide)

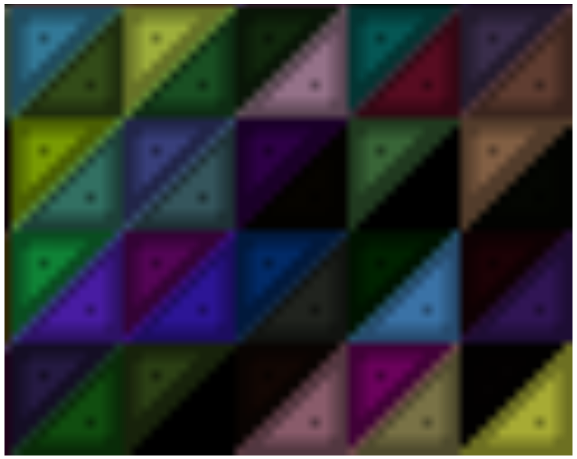
3D Vertex Coordinates

Triangular Facets/faces (with indices)

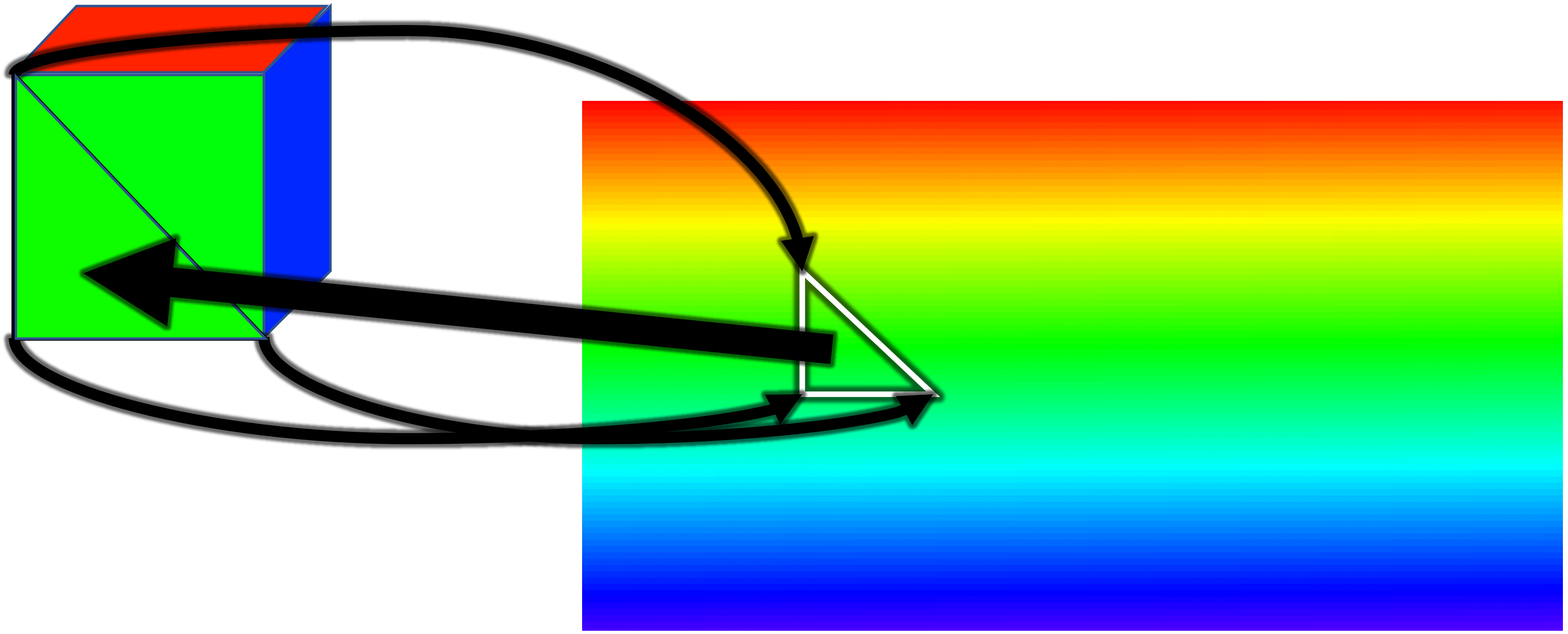
Texture Map (PNG/TGA) Files

- Surface (fill) colours are exported as a ***texture map***
- All surfaces are combined into a single image file
- Each surface is mapped to a region of that image

- A little weird, but it is done for efficiency
- It’s a bit like a sprite sheet in a game engine



Texture Maps



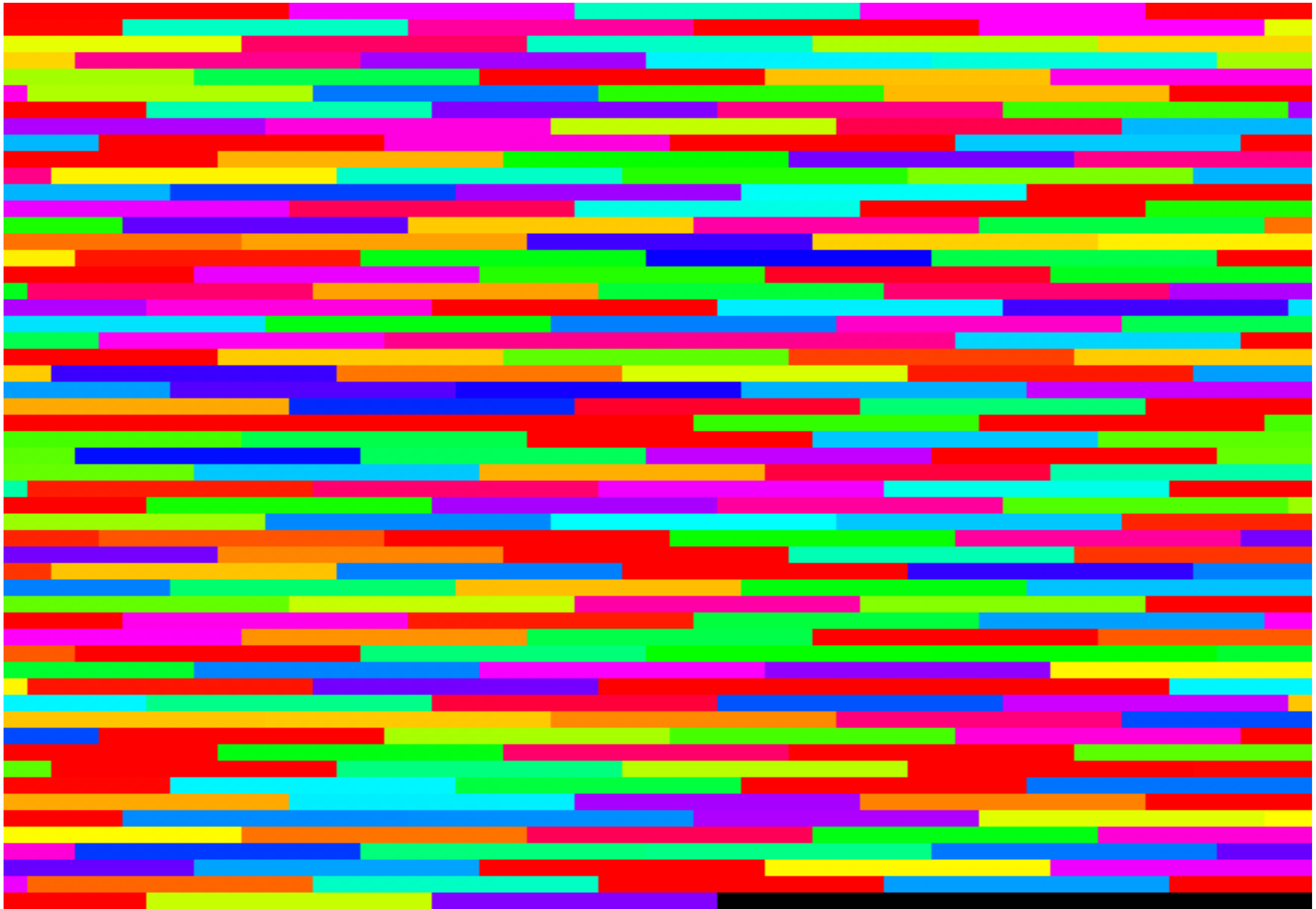
Things to watch out for !

- Texture map proportions can sometimes be > 1.0
- No idea why – seems a bit lazy to me !
- Just wrap around with modulus (%) to fix this

- Watch out for a flipped Y axis !
- Origin is sometimes the bottom left hand corner !!!

[FlyThrough – with Export]

Example Texture Map



MeshLab

- Open Source 3D model manipulation tool

  http://www.meshlab.net/

- Useful for testing your input/output
- Use MeshLab import to validate files…
- Files you generate (to make sure they are good !)
- Files you are trying to load (in case they are bad !)

Dynamic Data

- Since we are dynamically generating models
- It is possible to pull in data from various sources
- And use this to parameterise the model

- Could either be a live feed or some historic data
- Either way we end up with a graphical visualisation

[CityScape]