

Blocking and Advanced Blocking ?

Pierre Aubert



Blocking Principle

Stencil : 3×3



Blocking Principle

Stencil : 3×3



Read 3 rows at a time



Blocking Principle

Stencil : 3×3



Read 3 rows at a time



L1d cache : 48kB
(mostly 32kB)

Blocking Principle

Stencil : 3×3



Read 3 rows at a time



L1d cache : 48kB
(mostly 32kB)

4 Tables : U, V, U', V'

Float : 4 Bytes

Blocking Principle

Stencil : 3x3



Read 3 rows at a time



L1d cache : 48kB
(mostly 32kB)

4 Tables : U, V, U', V'

Float : 4 Bytes

3000 Floats per block

At least
1000 Floats per block
row

Blocking Principle

Stencil : 3×3



Read 3 rows at a time



L1d cache : 48kB
(mostly 32kB)

4 Tables : U, V, U', V'

Float : 4 Bytes

3000 Floats per block

At least
1000 Floats per block
row

But HD image : 1920×1080

Blocking Principle

Stencil : 3×3



Read 3 rows at a time



L1d cache : 48kB
(mostly 32kB)

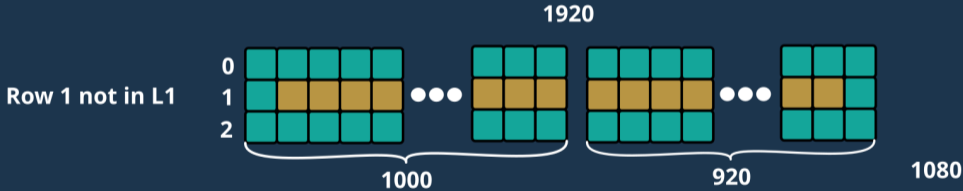
4 Tables : U, V, U', V'

Float : 4 Bytes

3000 Floats per block

At least
1000 Floats per block
row

But HD image : 1920×1080



Blocking Principle

Stencil : 3×3



Read 3 rows at a time



L1d cache : 48kB
(mostly 32kB)

4 Tables : U, V, U', V'

Float : 4 Bytes

3000 Floats per block

At least
1000 Floats per block
row

But HD image : 1920×1080

1920

Row 1 not in L1



Reload Row 1
from L2



Blocking Principle

Blocking Principle

Block => Reusing rows in L1d

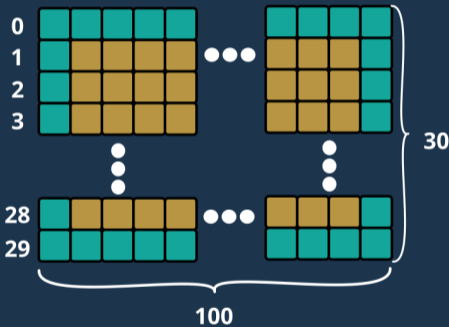
Still 3000 floats per block

Blocking Principle

Block => Reusing rows in L1d

Still 3000 floats per block

Example block : 30x100

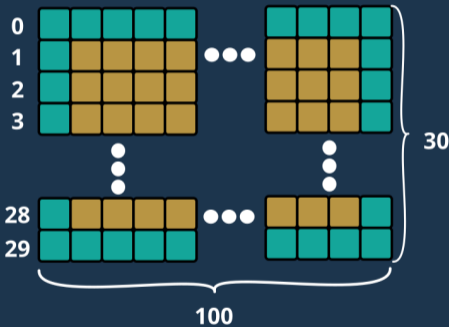


Blocking Principle

Block => Reusing rows in L1d

Still 3000 floats per block

Example block : 30×100



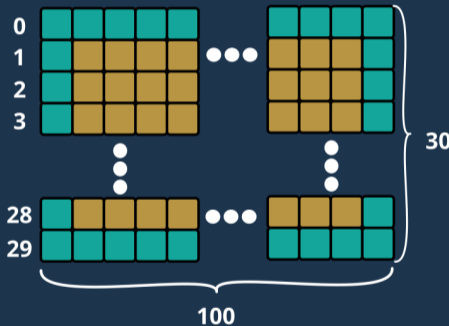
Previous row always in L1d cache

Blocking Principle

Block => Reusing rows in L1d

Still 3000 floats per block

Example block : 30×100



Previous row always in L1d cache

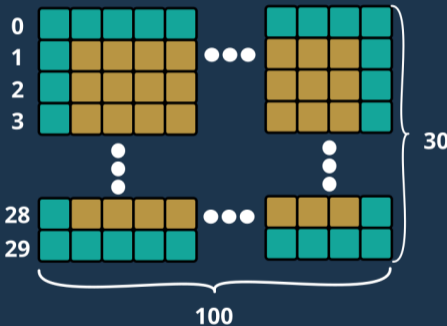
Blocks have to overlap to compute borders correctly

Blocking Principle

Block => Reusing rows in L1d

Still 3000 floats per block

Example block : 30x100



Previous row always in L1d cache

Blocks have to overlap to compute borders correctly

No Block

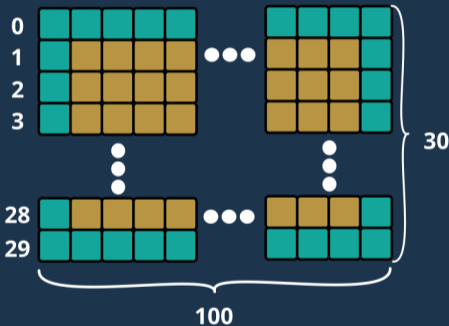
3 reads from L2

Blocking Principle

Block => Reusing rows in L1d

Still 3000 floats per block

Example block : 30x100

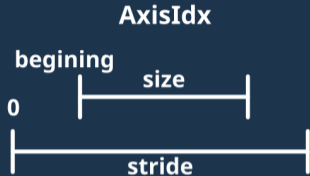


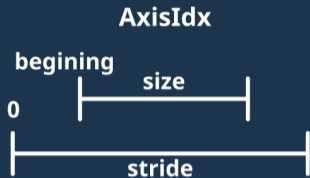
Previous row always in L1d cache

Blocks have to overlap to compute borders correctly

No Block
3 reads from L2

With block
3 reads :
- 1 from L2
- 2 from L1d



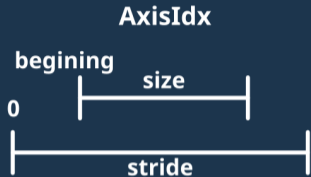


BlockIdx

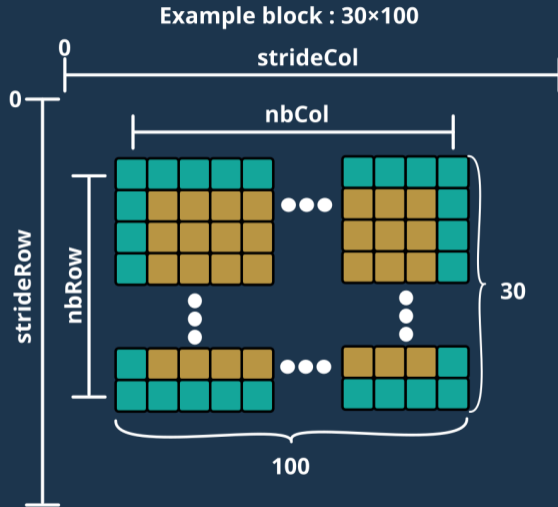
AxisIdx row

AxisIdx col

AxisIdx and BlockIdx

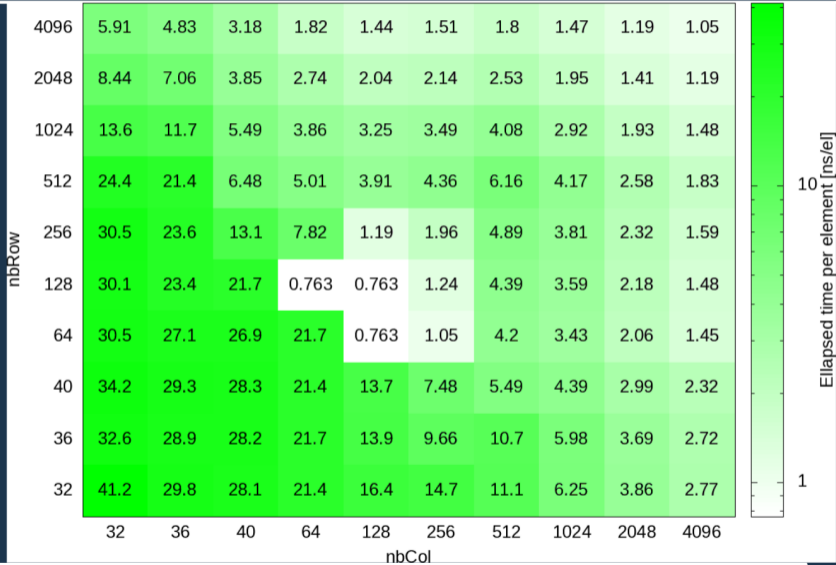


BlockIdx
AxisIdx row
AxisIdx col



Best Block Size/Shape ?

Block



Elapsed time per element [ns/el]

1

10

Block



For
Gray Scott
Simulation

In ns/el

Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)

t = 1



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Blocking Pyramid 1D

Each iteration spreads information by one pixel

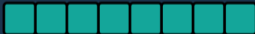
Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Block (8 cells)



Same amount of data needed
than classical block

Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Block (8 cells)



Blocking Pyramid 1D

Each iteration spreads information by one pixel

Take advantage of extra steps

Block (8 cells)



Inner values
not needed

Block (8 cells)



Same amount of data needed
than classical block

Block (8 cells)



Block Classic (8 cells)

Block Classic (8 cells)



Image 1



Block Classic (8 cells)



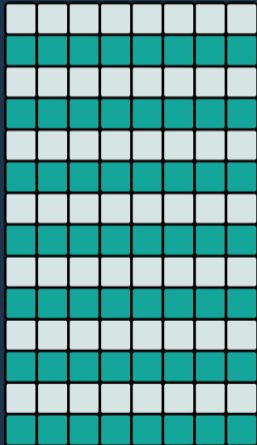
Image 1

Image 2



Block Classic (8 cells)

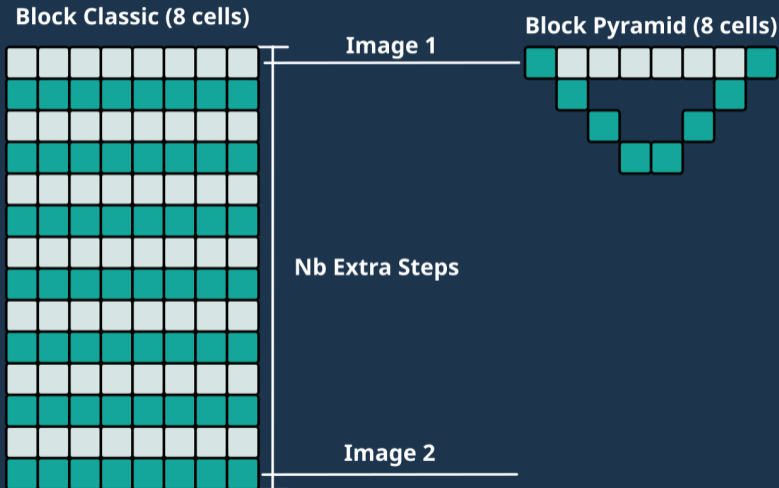
Image 1



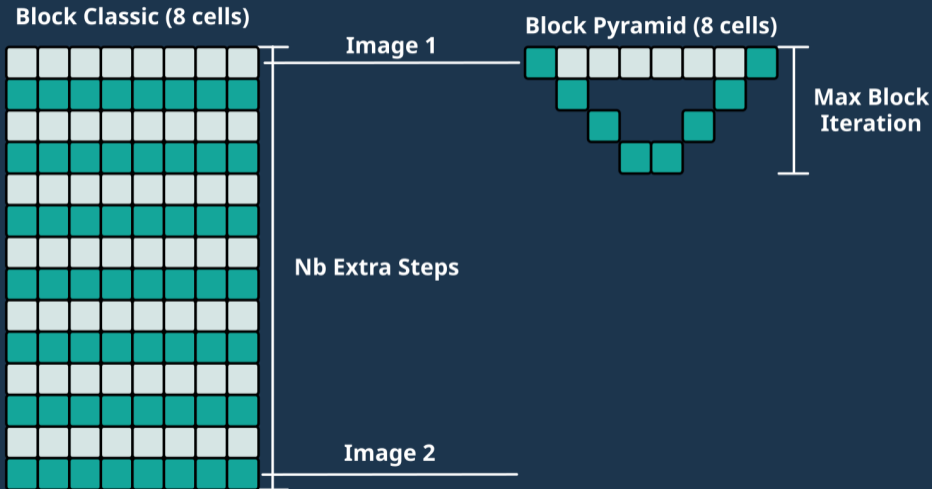
Nb Extra Steps

Image 2

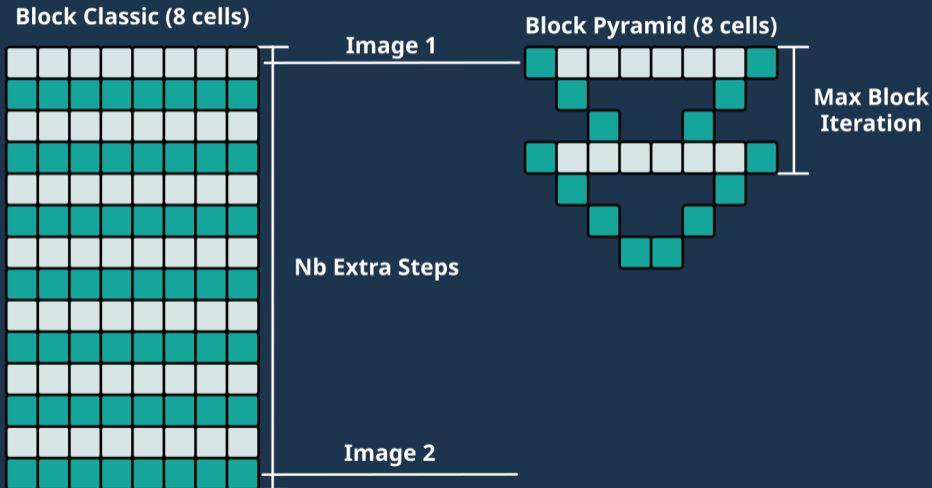
Pyramid Iteration 1D



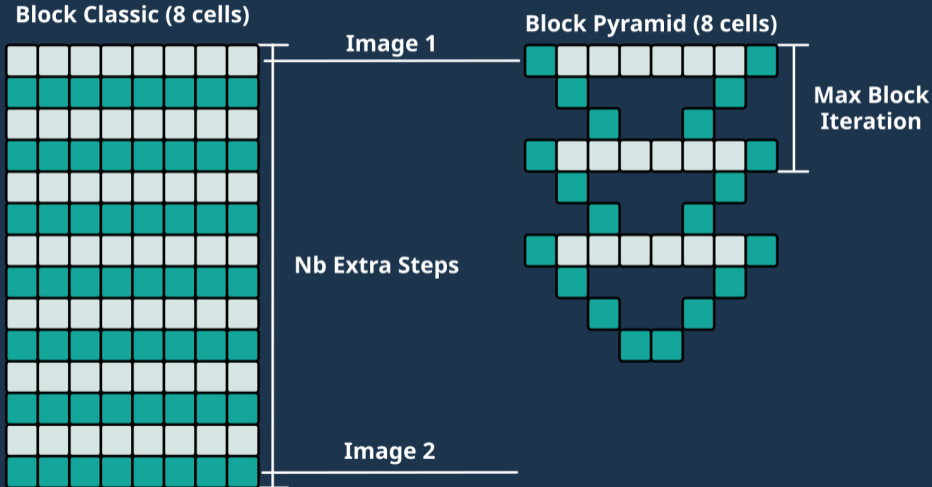
Pyramid Iteration 1D



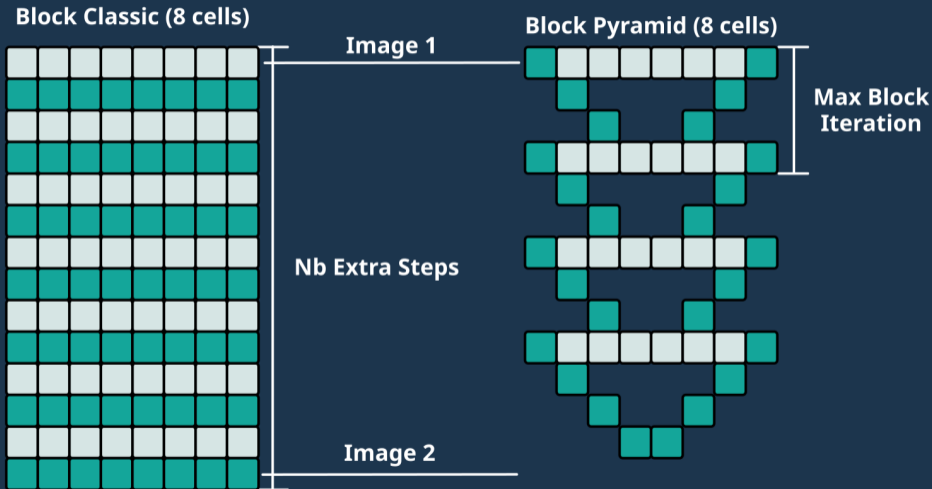
Pyramid Iteration 1D



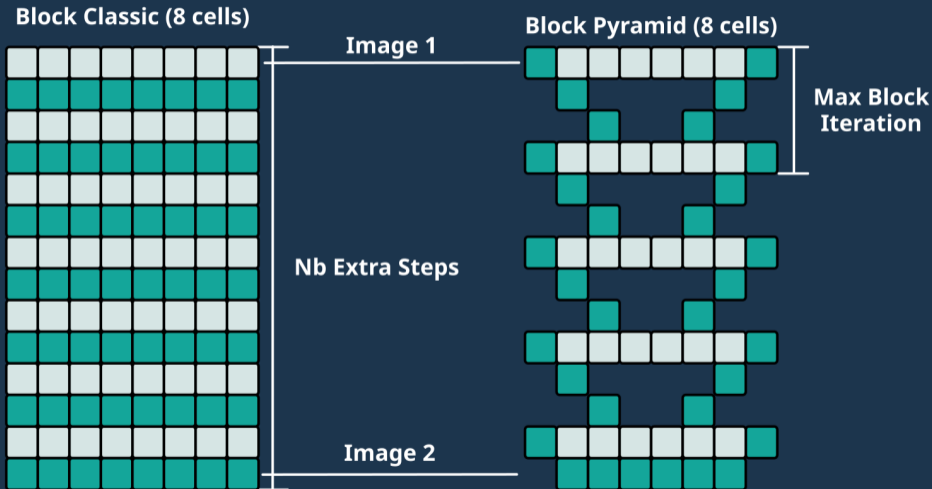
Pyramid Iteration 1D



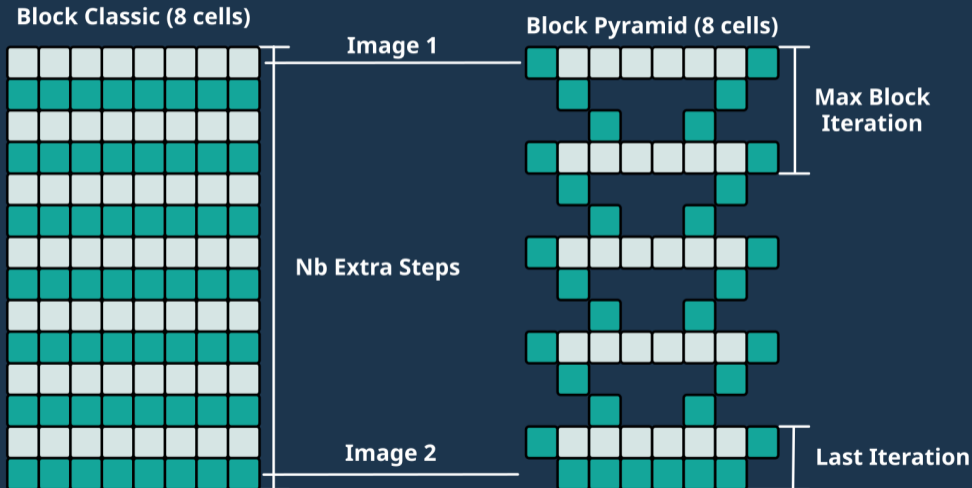
Pyramid Iteration 1D



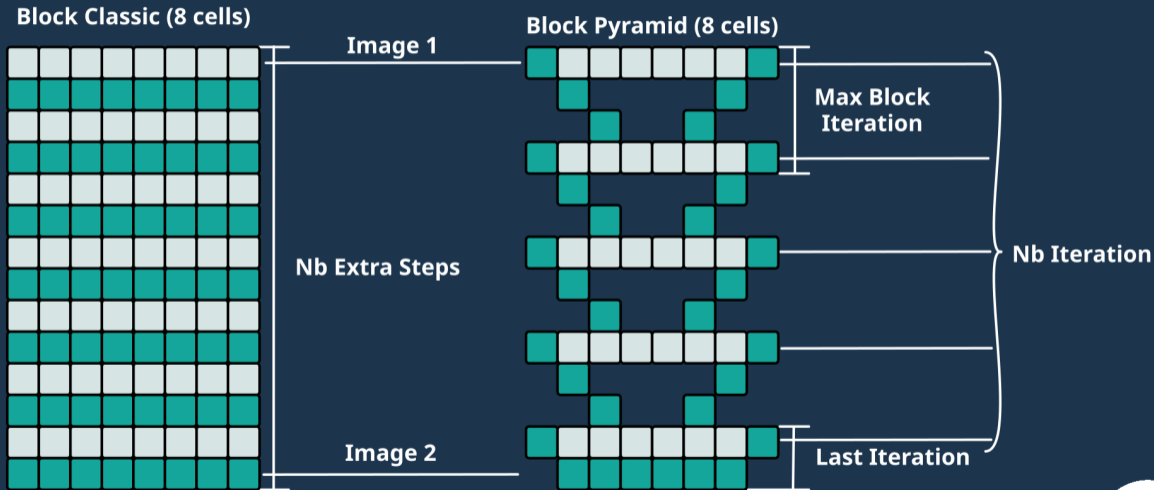
Pyramid Iteration 1D



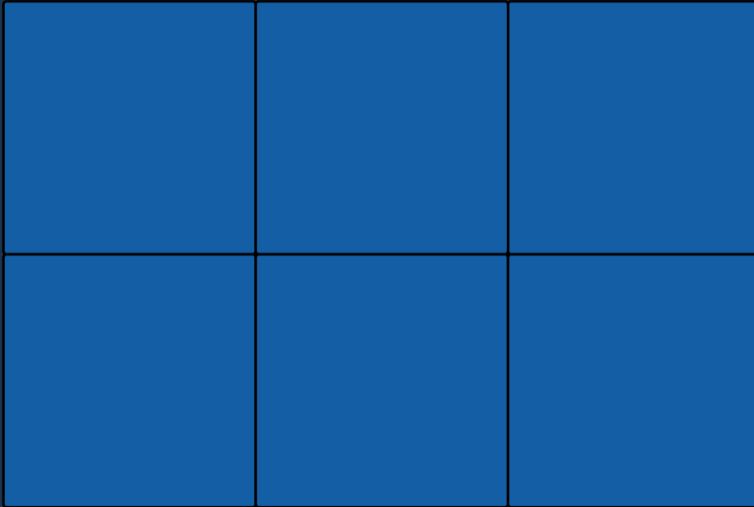
Pyramid Iteration 1D



Pyramid Iteration 1D



Blocking Pyramid

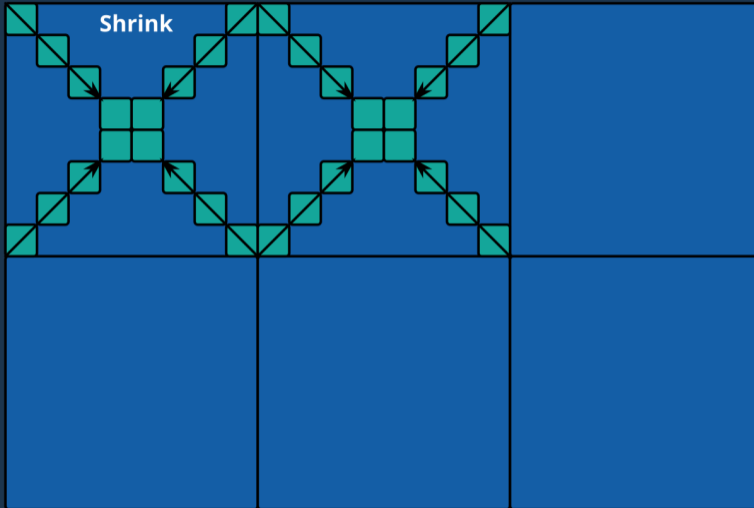


Blocking Pyramid



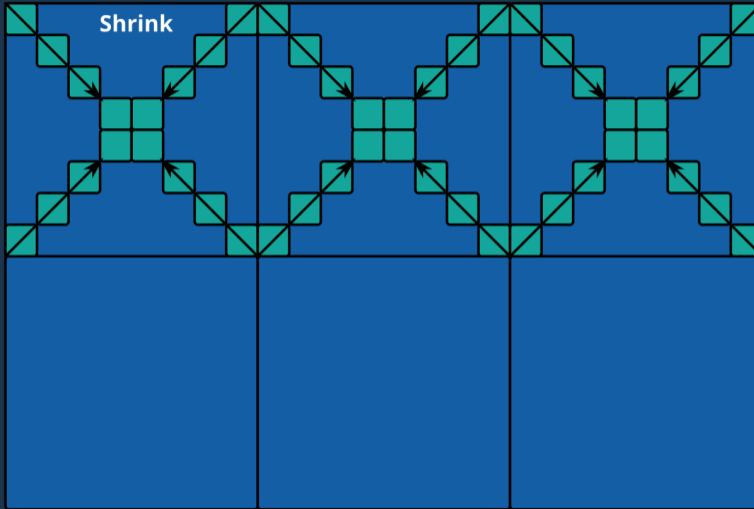
Starting with blue blocks

Blocking Pyramid



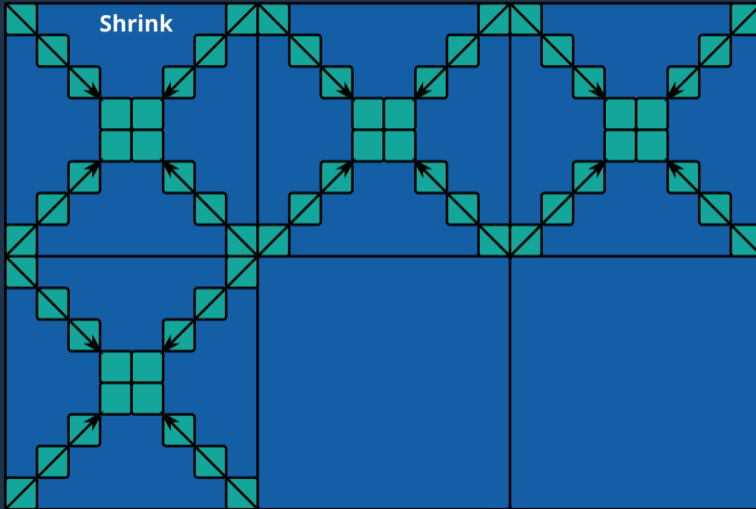
Starting with blue blocks

Blocking Pyramid



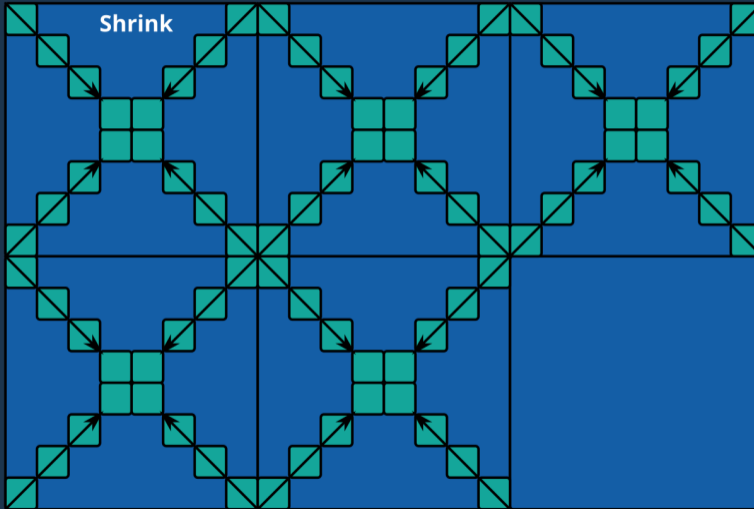
Starting with blue blocks

Blocking Pyramid



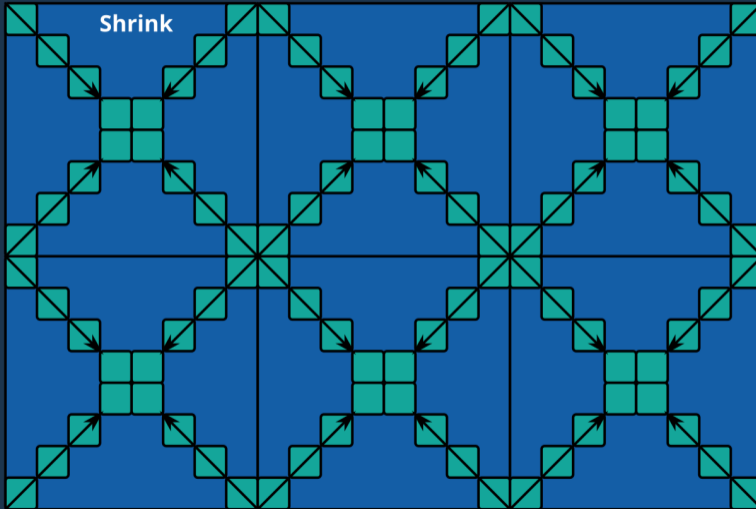
Starting with blue blocks

Blocking Pyramid



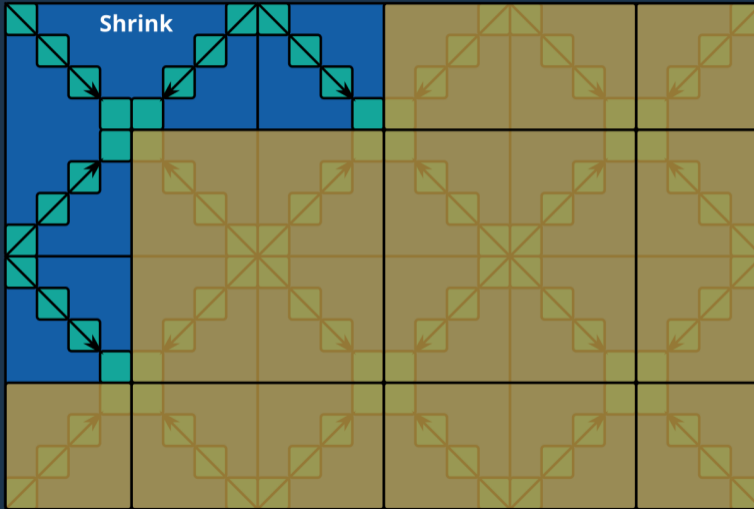
Starting with blue blocks

Blocking Pyramid



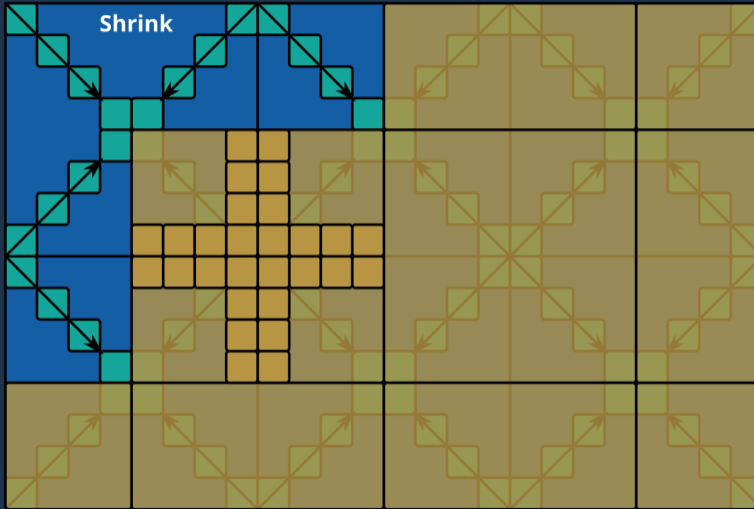
Starting with blue blocks

Blocking Pyramid



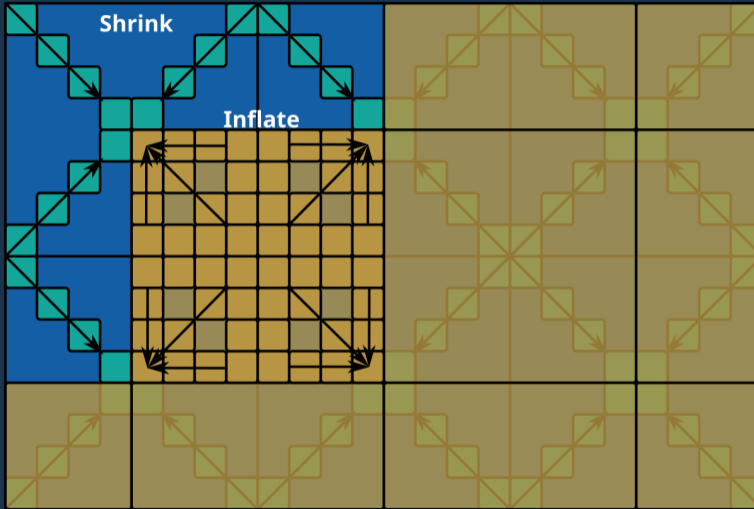
Starting with blue blocks

Blocking Pyramid



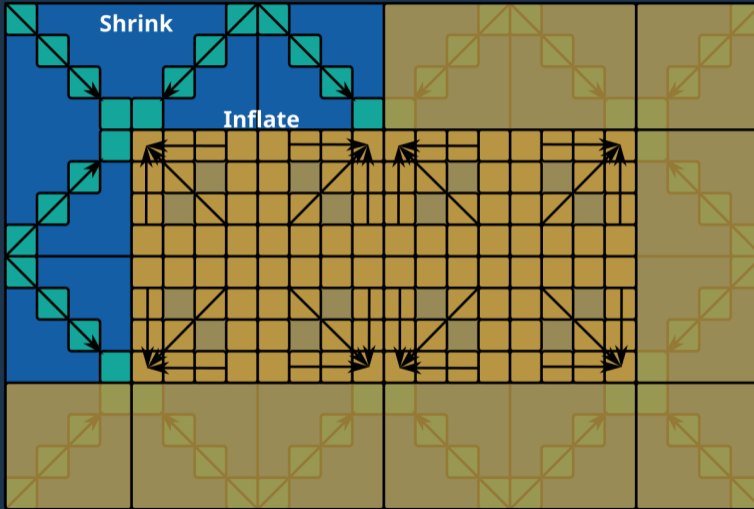
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



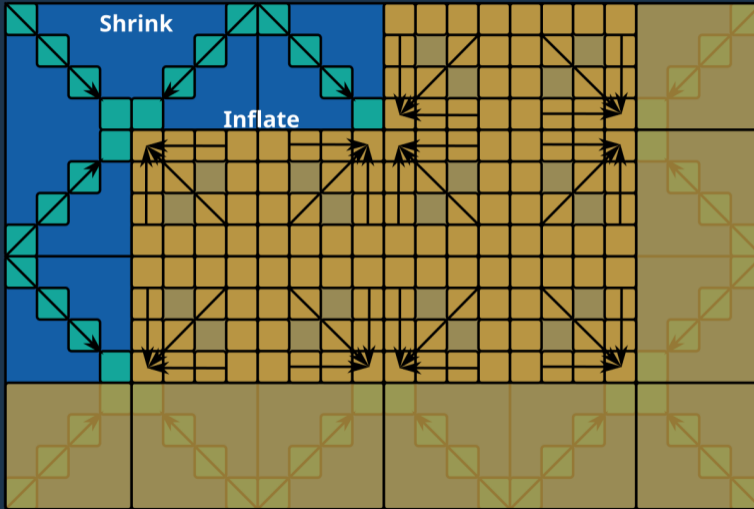
-  Starting with blue blocks
-  Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



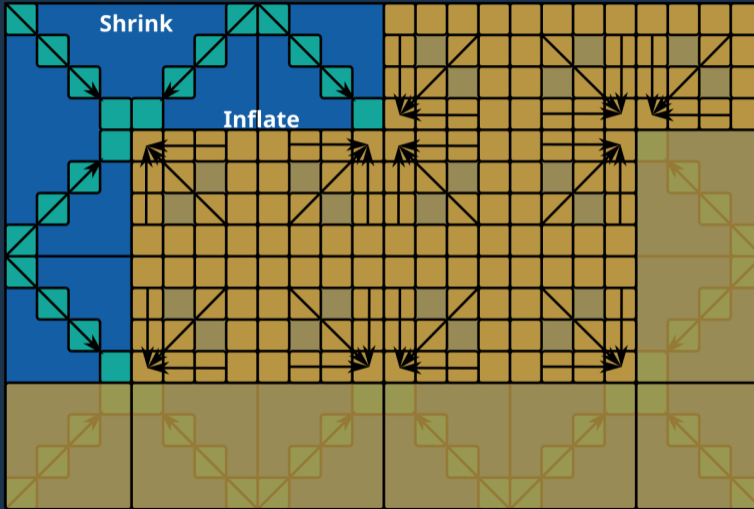
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



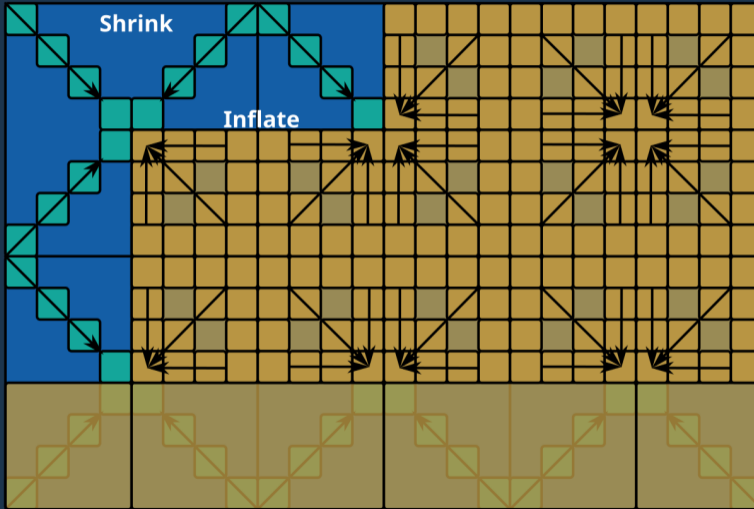
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



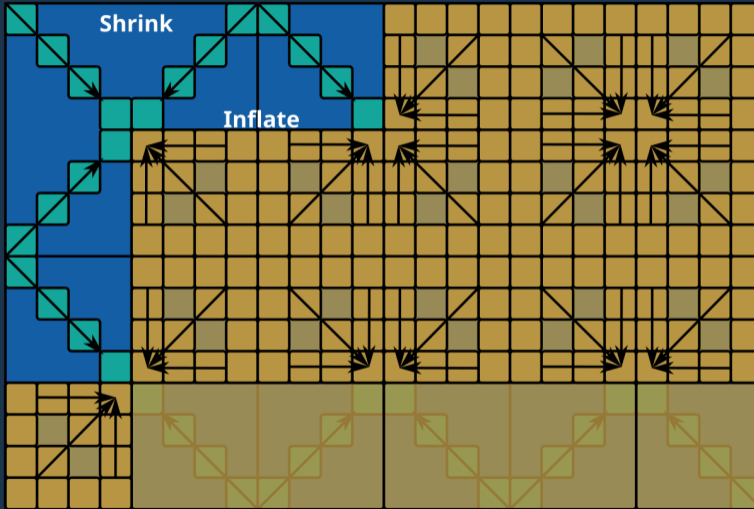
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



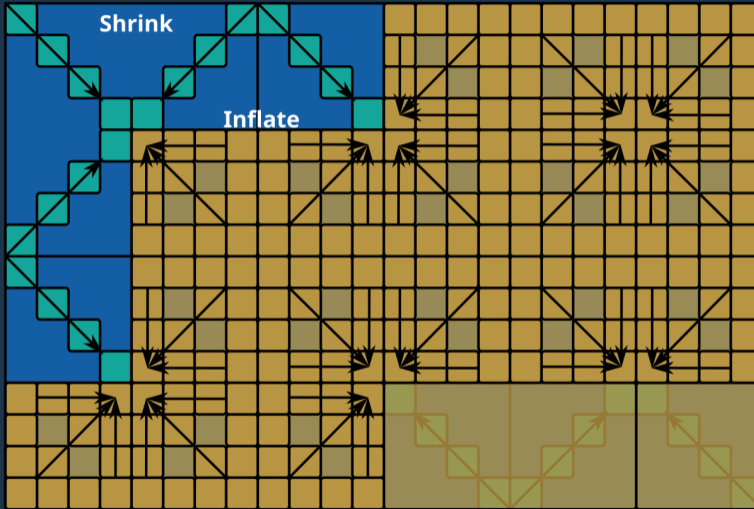
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



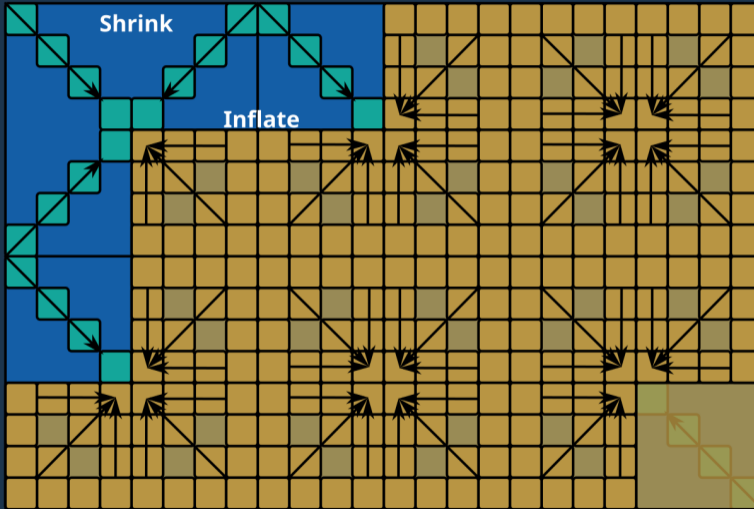
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



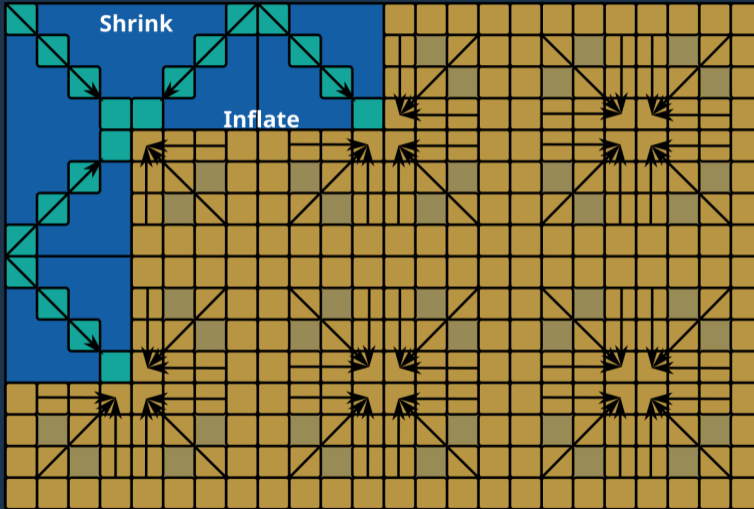
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



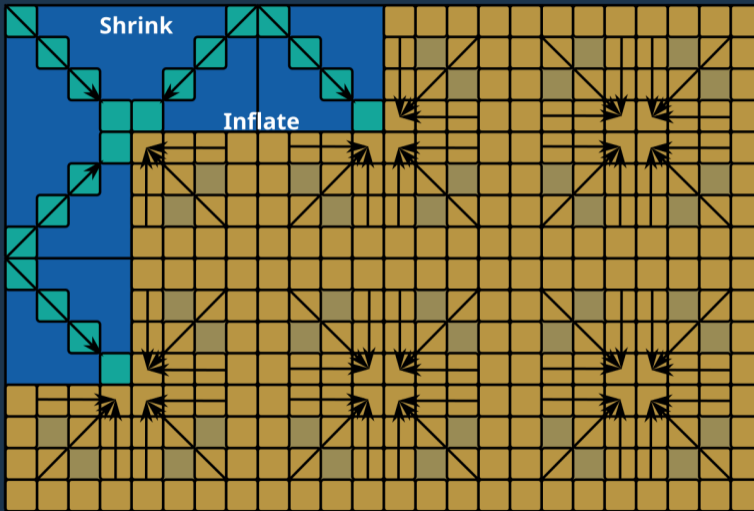
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

Blocking Pyramid



- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

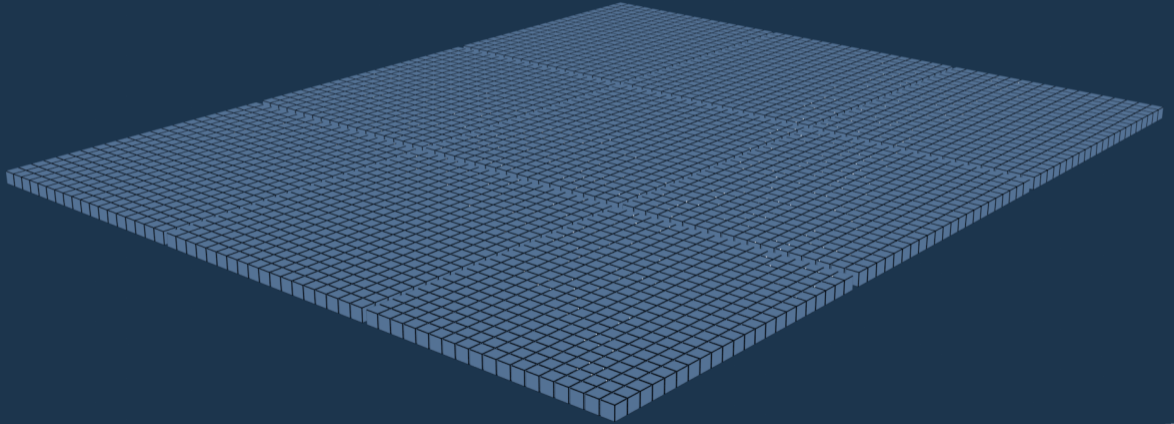
Blocking Pyramid



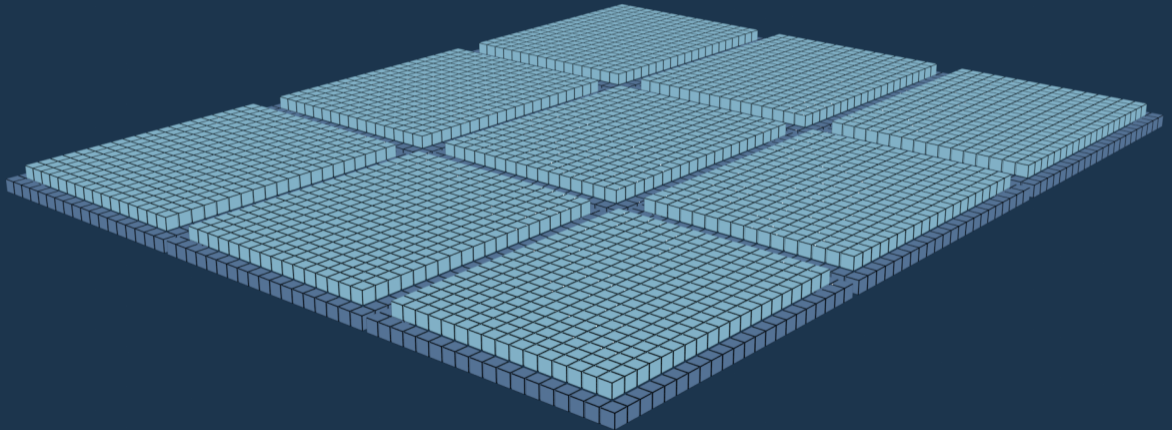
- Starting with blue blocks
- Propagate the rest of the pixel with yellow blocks

No explicit communication needed between blocks on the same machine

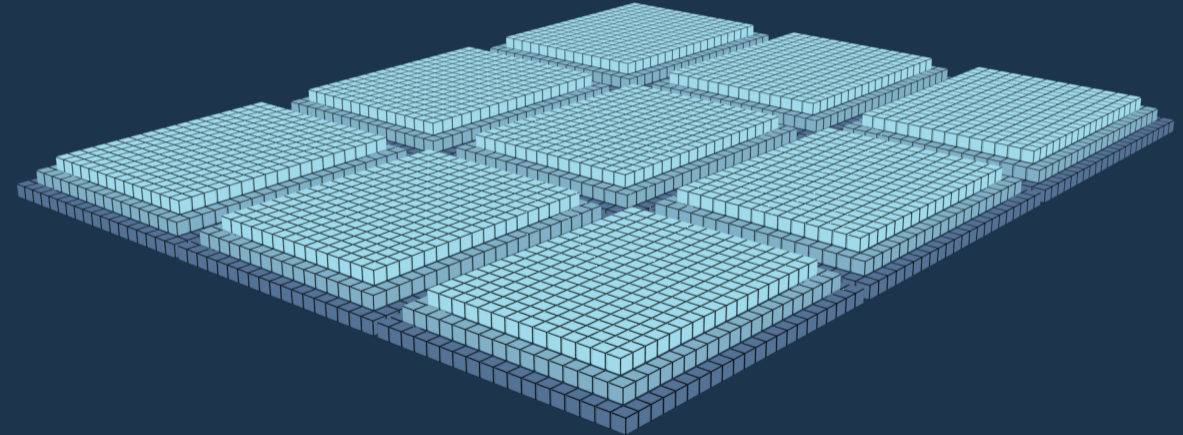
Blocking Pyramid



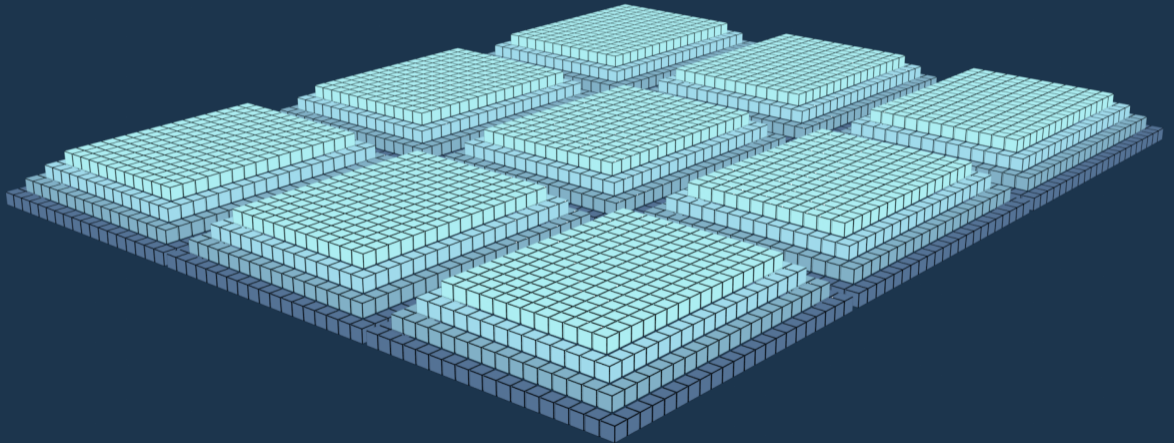
Blocking Pyramid



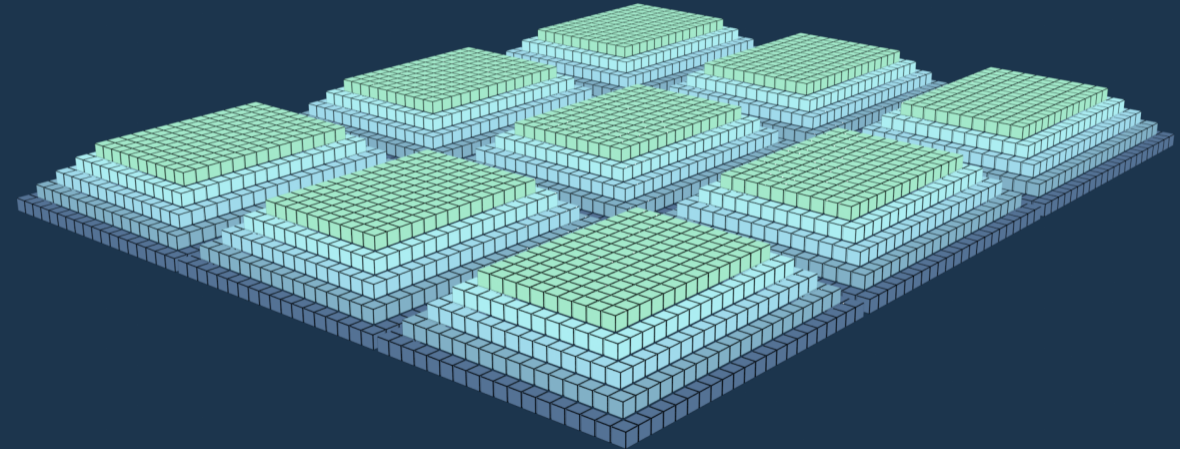
Blocking Pyramid



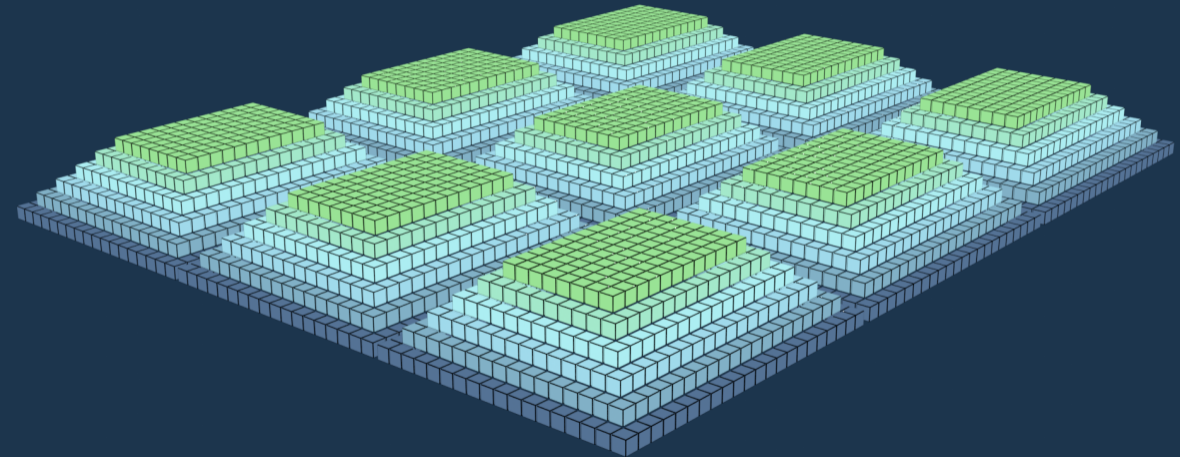
Blocking Pyramid



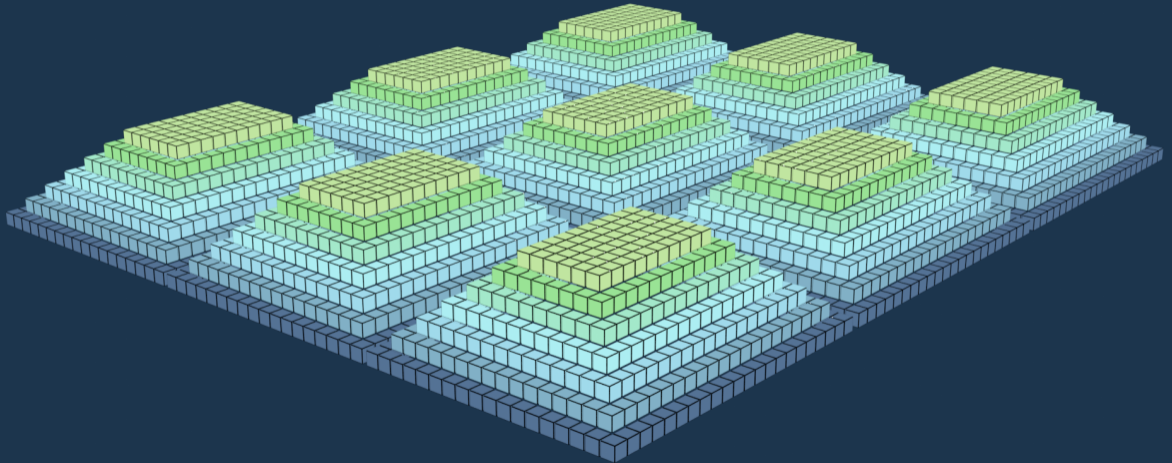
Blocking Pyramid



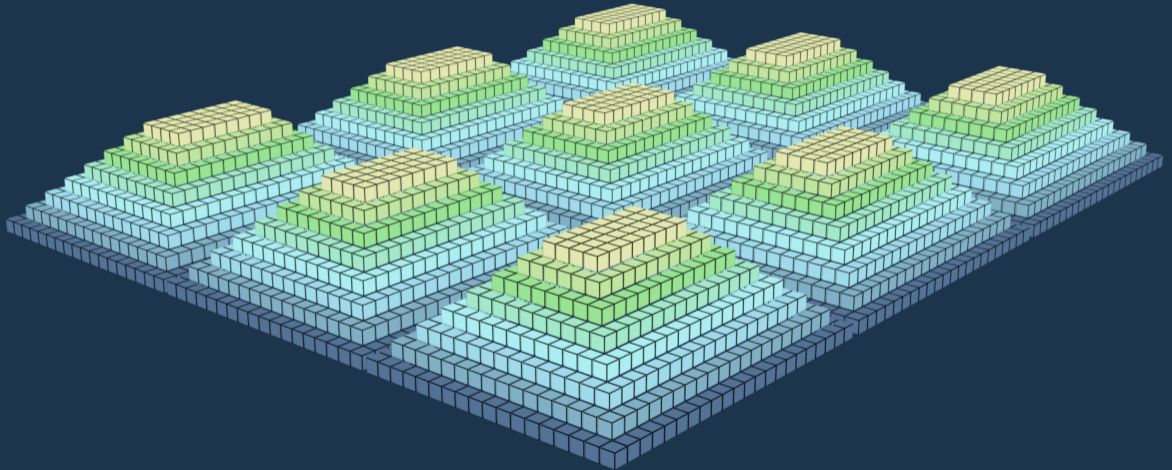
Blocking Pyramid



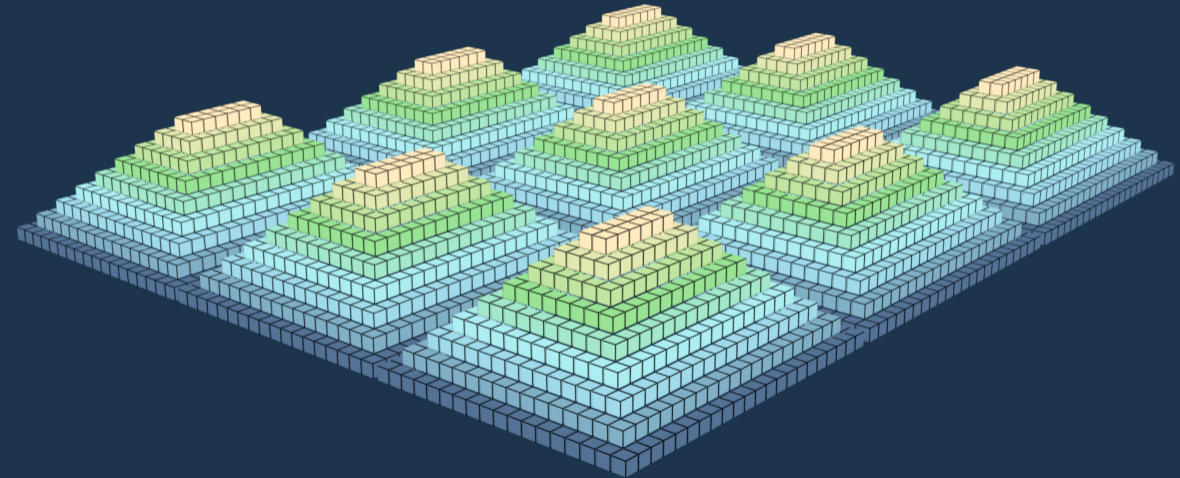
Blocking Pyramid



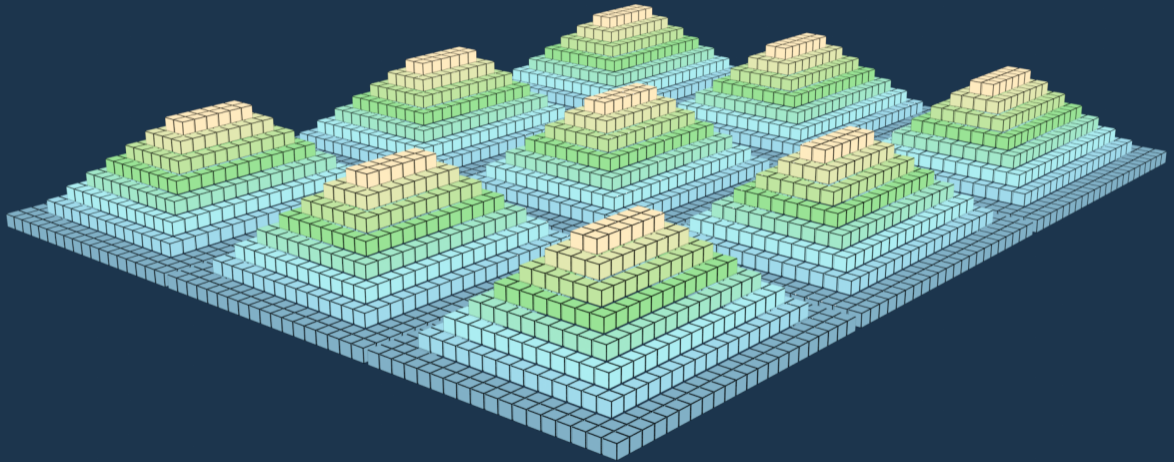
Blocking Pyramid



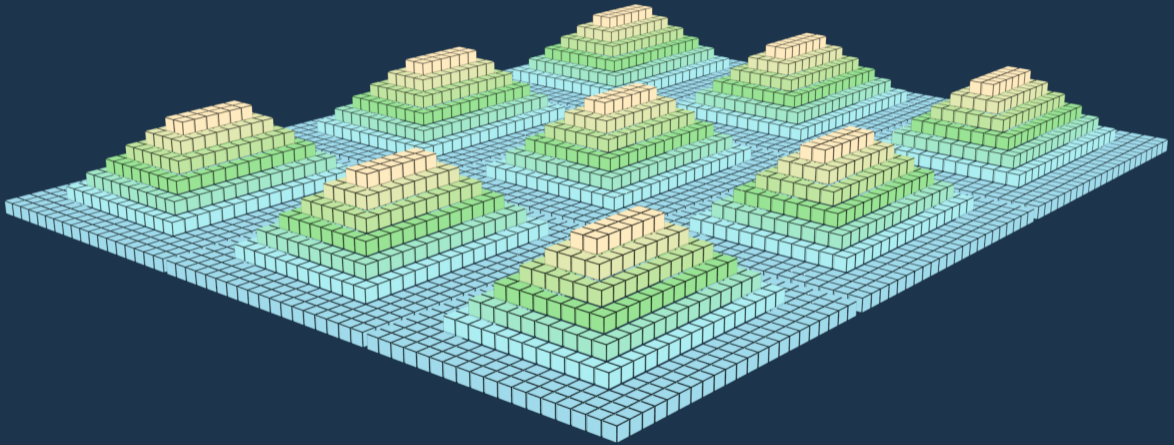
Blocking Pyramid



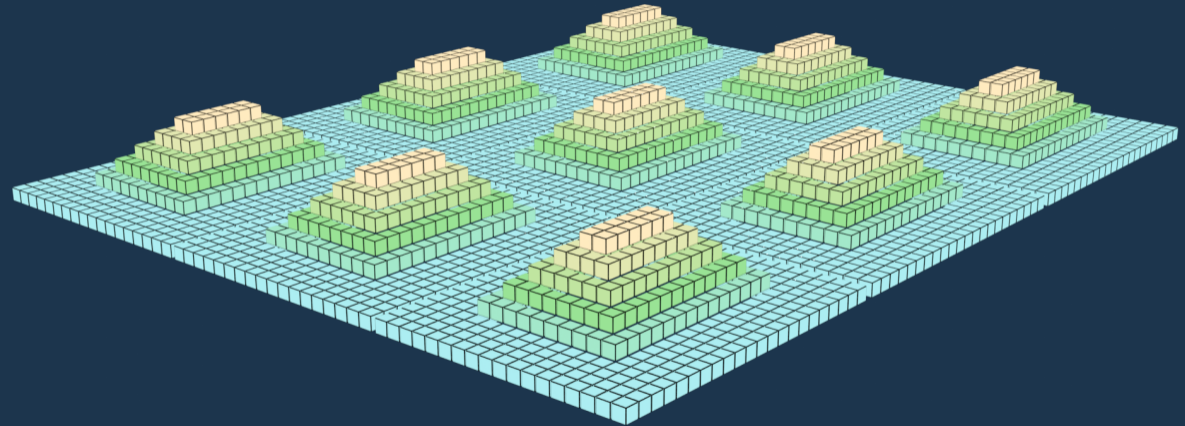
Blocking Pyramid



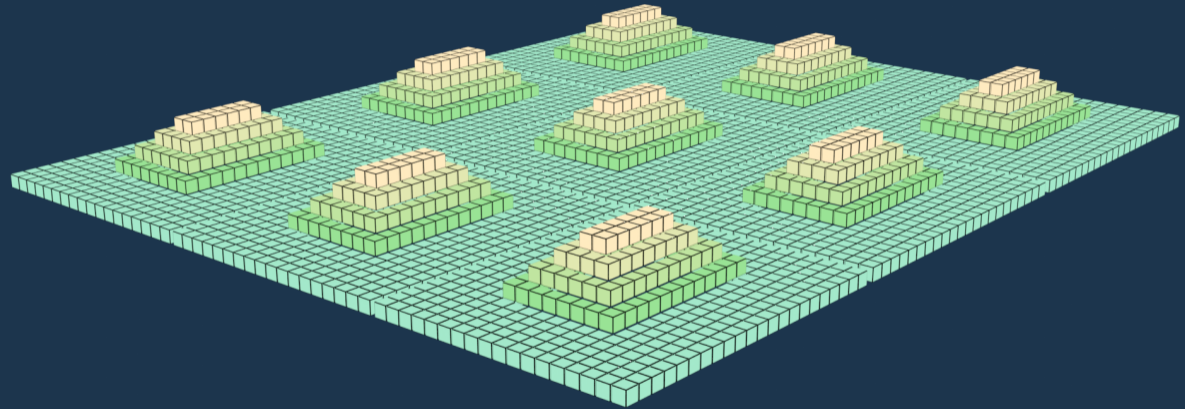
Blocking Pyramid



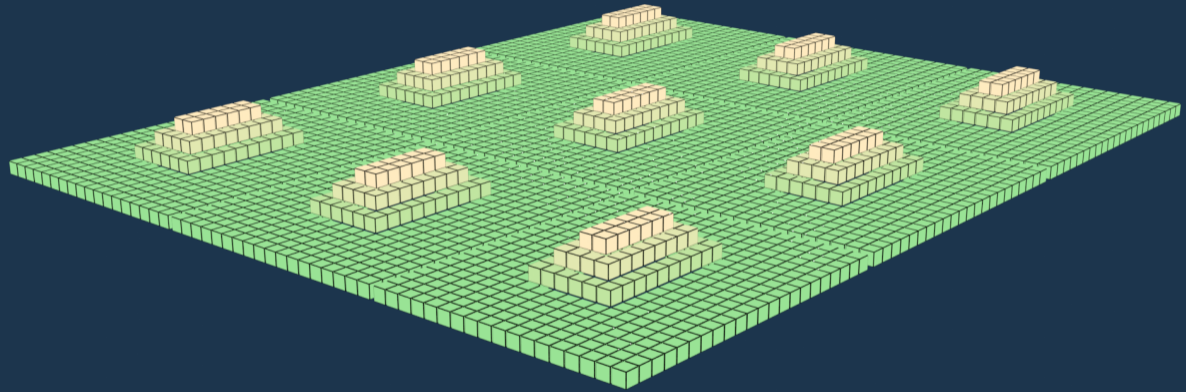
Blocking Pyramid



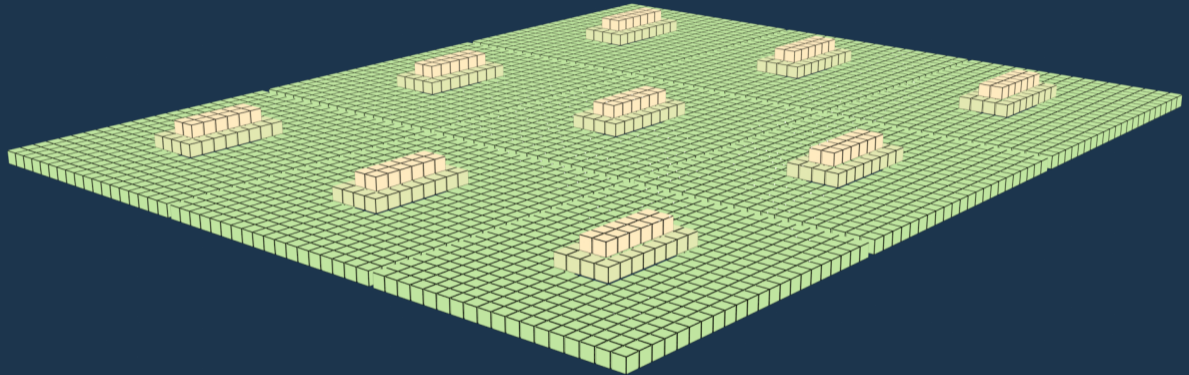
Blocking Pyramid



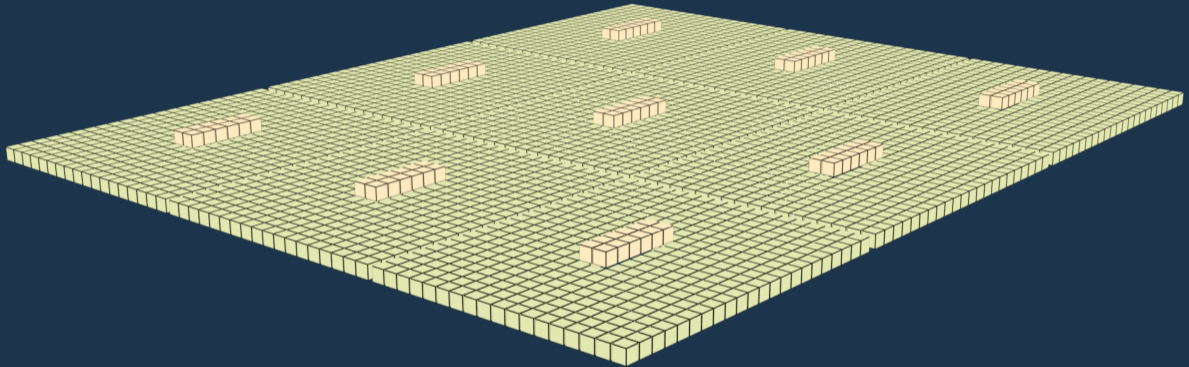
Blocking Pyramid



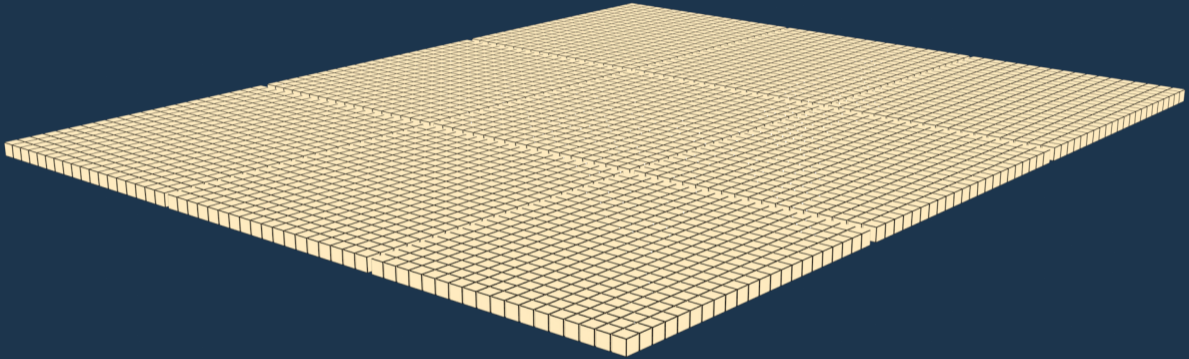
Blocking Pyramid



Blocking Pyramid



Blocking Pyramid

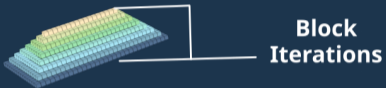


Pyramid Parameters ?

Pyramid Parameters ?

L1 Cache

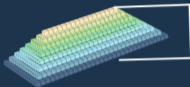
- Smaller Pyramid
- Less Block Iterations
- Faster Data Access



Pyramid Parameters ?

L1 Cache

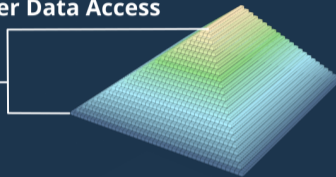
- Smaller Pyramid
- Less Block Iterations
- Faster Data Access



Block
Iterations

L2 Cache

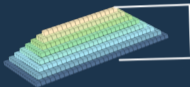
- Bigger Pyramid
- More Block Iterations
- Slower Data Access



Pyramid Parameters ?

L1 Cache

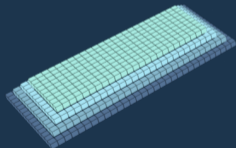
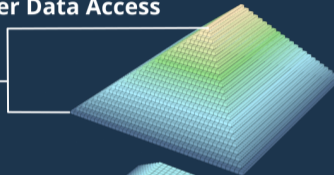
- Smaller Pyramid
- Less Block Iterations
- Faster Data Access



Block
Iterations

L2 Cache

- Bigger Pyramid
- More Block Iterations
- Slower Data Access



Reduce block
iterations to keep
efficient
vectorization

