Laboratoire d'Annecy de Physique des Particules

Simpler Advanced Blocking ?

Pierre Aubert

ESCAPE

cta cherenkov telescope array

cnrs

UNIVERSITÉ SAVOIE MONT BLANC

Simpler Pyramid

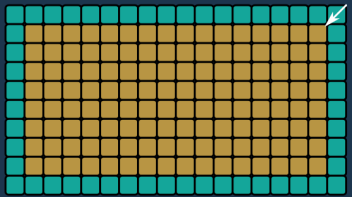
Simpler Pyramid

3x3 Stencil 

Simpler Pyramid

3x3 Stencil 

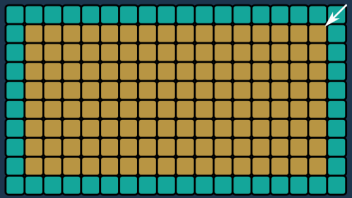
1 step in the future



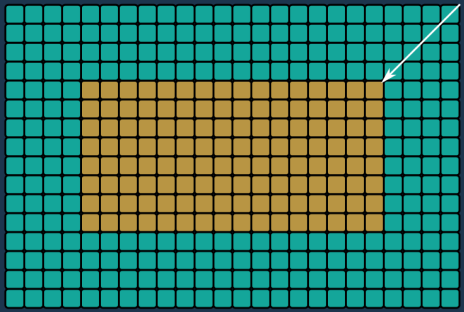
Simpler Pyramid

3x3 Stencil 

1 step in the future



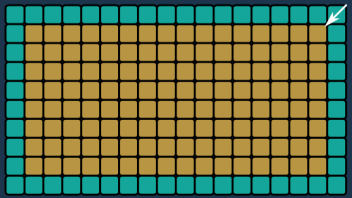
4 steps in the future



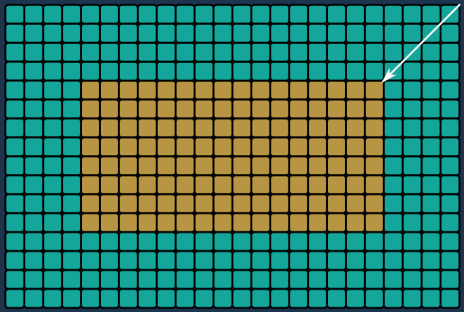
Simpler Pyramid

3x3 Stencil 

1 step in the future



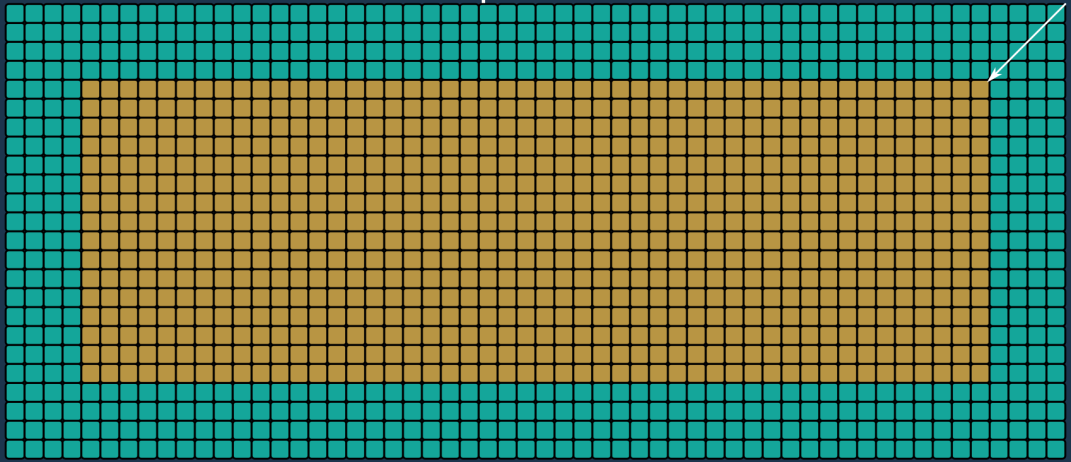
4 steps in the future



Blocks have to be large enough

Simpler Pyramid

4 steps in the future

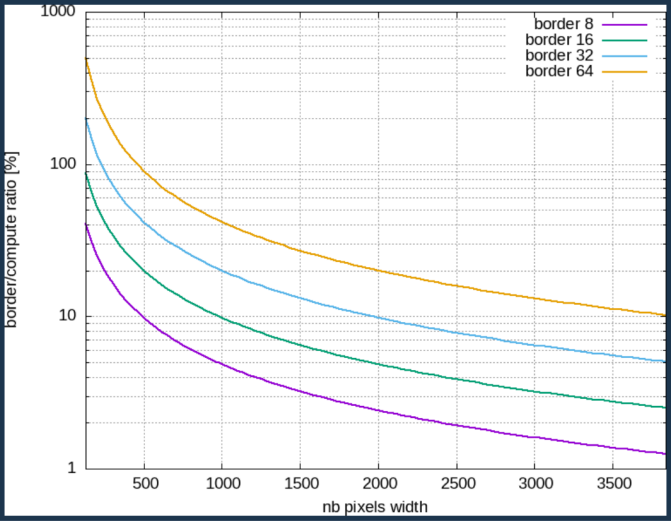


Compute: 16×48 = 768 pixels

Border: 2×4×24 + 2×4×48 = 576 pixels

Ratio (border/compute): 75%

Simpler Pyramid

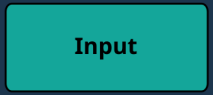


Here:
height = width/2

Distributed Simpler Pyramid

Distributed Simpler Pyramid

Attributes
- currentImage
- deviceType (CPU, GPU)



Distributed Simpler Pyramid

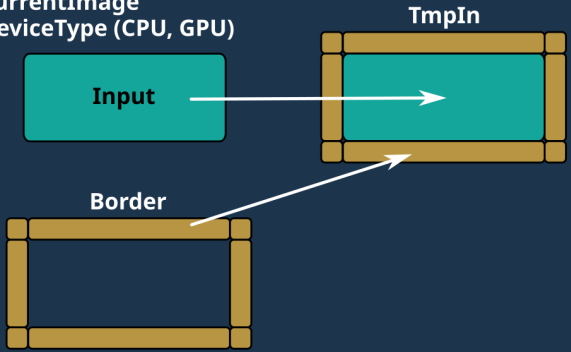
Attributes
- currentImage
- deviceType (CPU, GPU)

Border

Waiting if neighbours completed their works before the current block

Distributed Simpler Pyramid

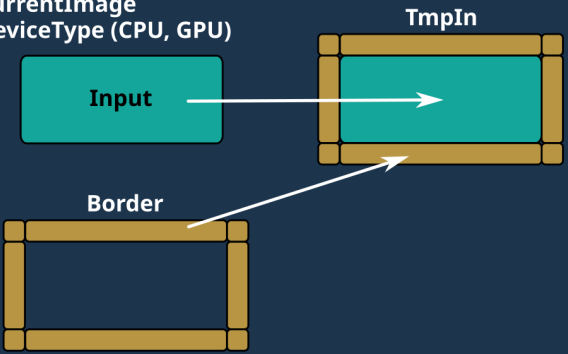
Attributes
- currentImage
- deviceType (CPU, GPU)



Waiting if neighbours completed their works before the current block

Distributed Simpler Pyramid

Attributes
- currentImage
- deviceType (CPU, GPU)



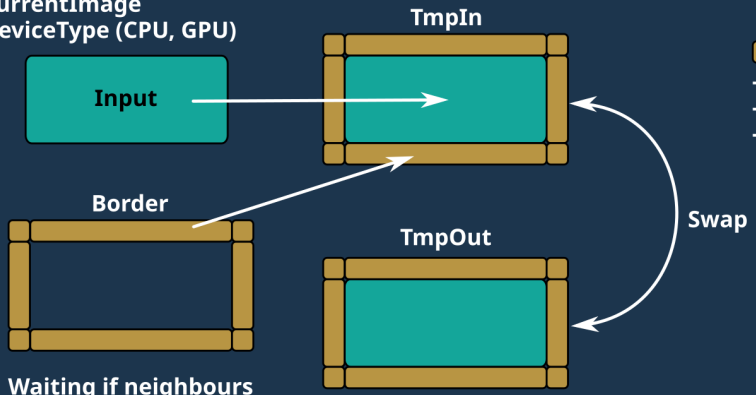
Waiting if neighbours completed their works before the current block

Border = Nb Pyramid Iterations

- offsetWidth
- offsetHeight
- currentImage

Distributed Simpler Pyramid

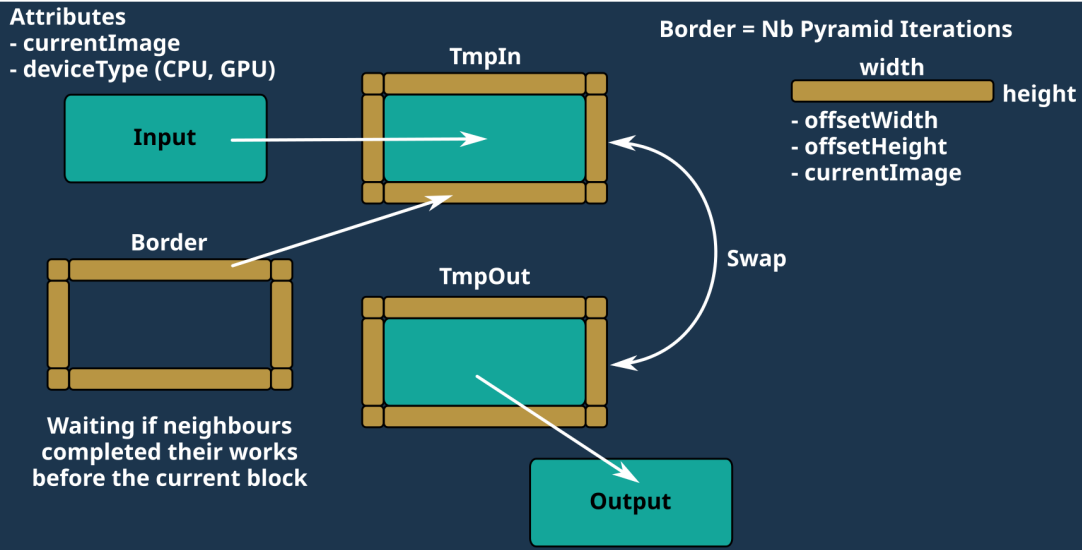
Attributes
- currentImage
- deviceType (CPU, GPU)



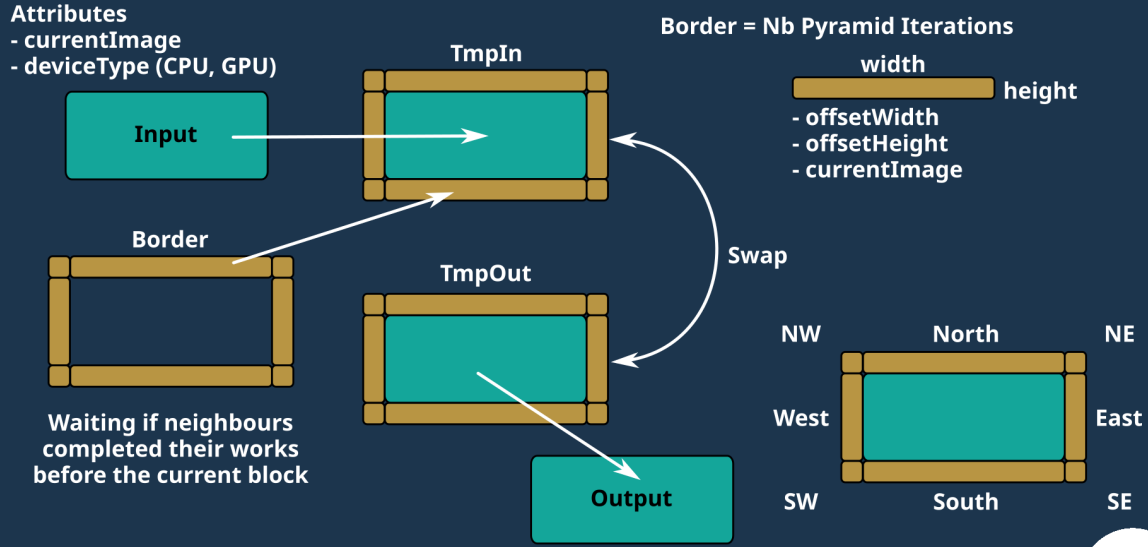
Border = Nb Pyramid Iterations

- offsetWidth
- offsetHeight
- currentImage

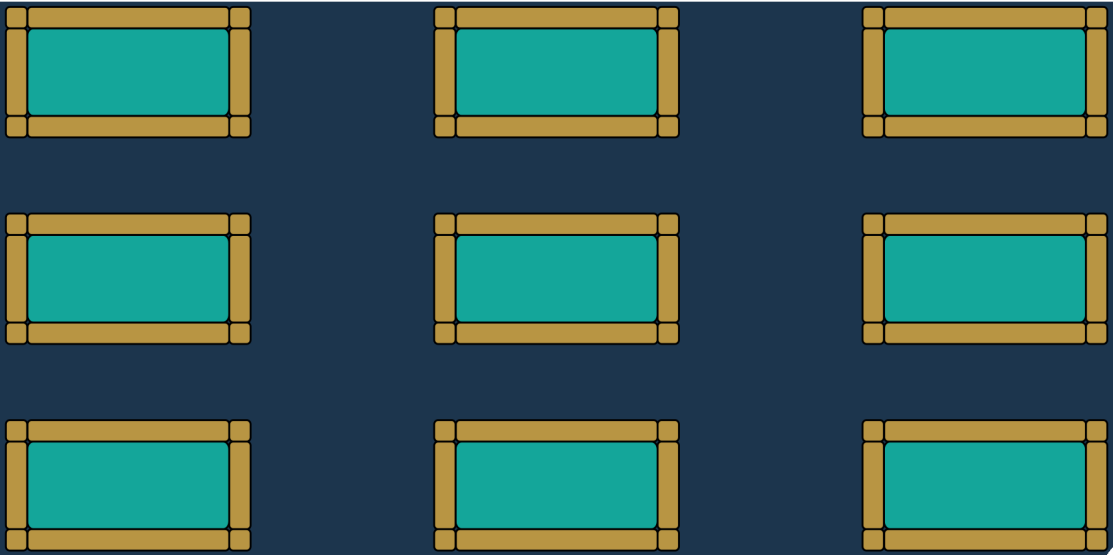
Distributed Simpler Pyramid



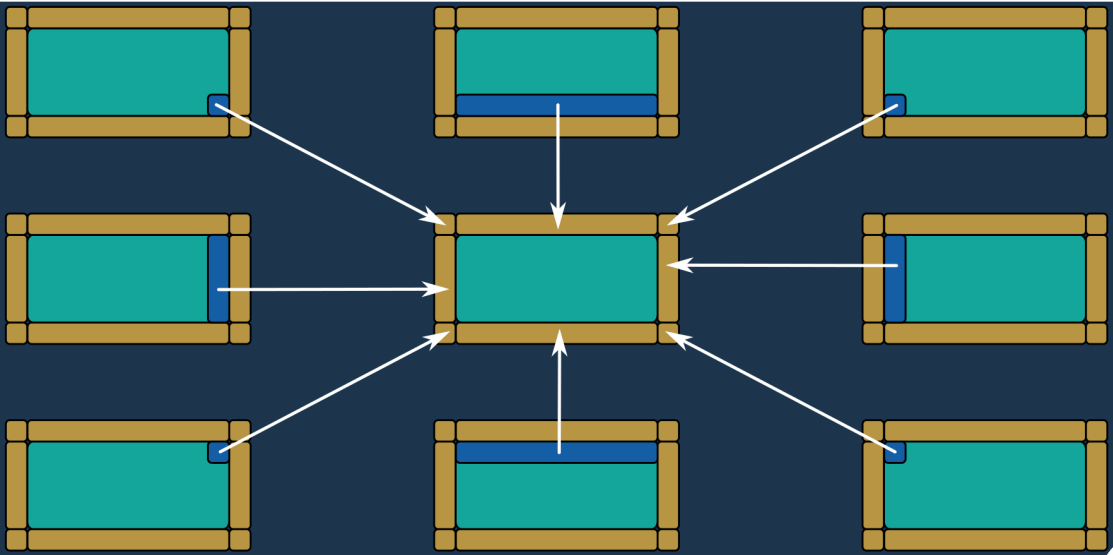
Distributed Simpler Pyramid



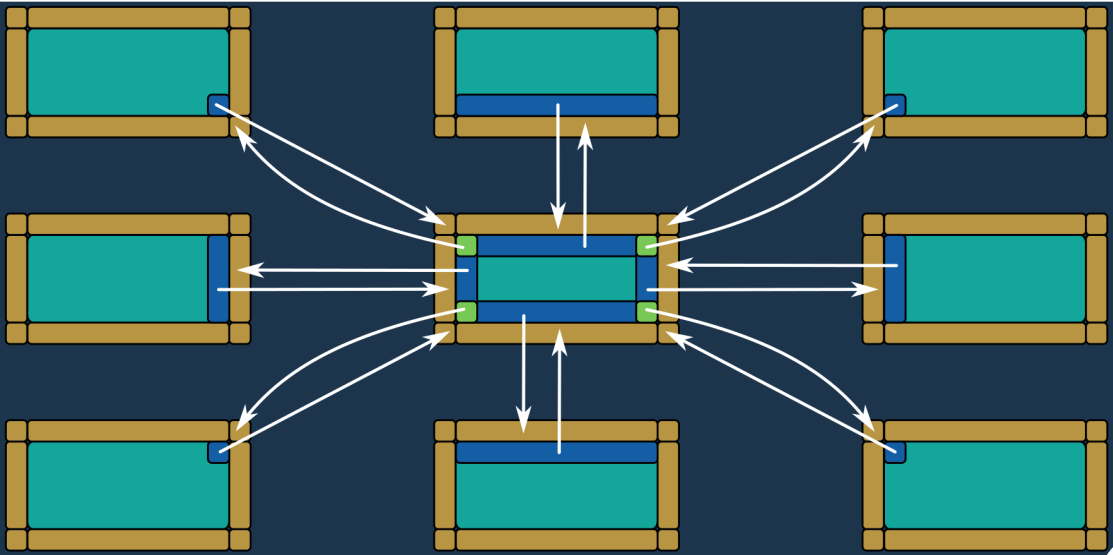
Distributed Simpler Pyramid



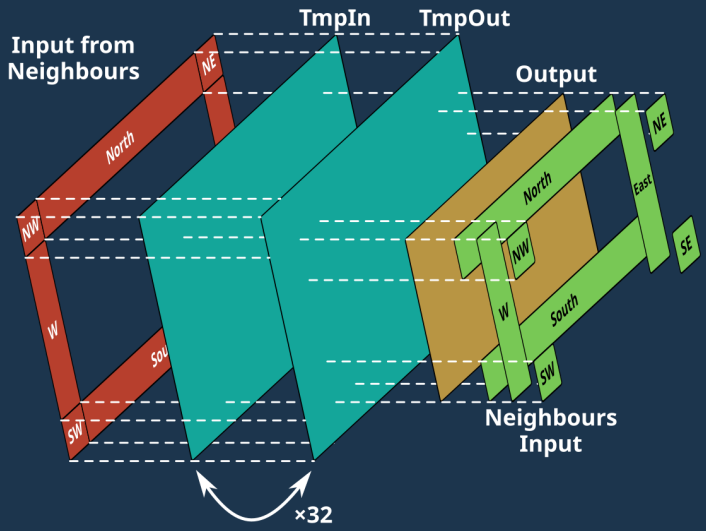
Distributed Simpler Pyramid



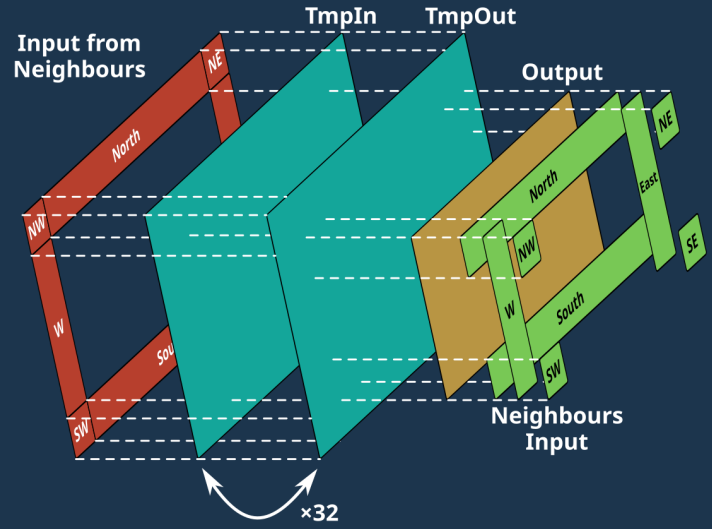
Distributed Simpler Pyramid



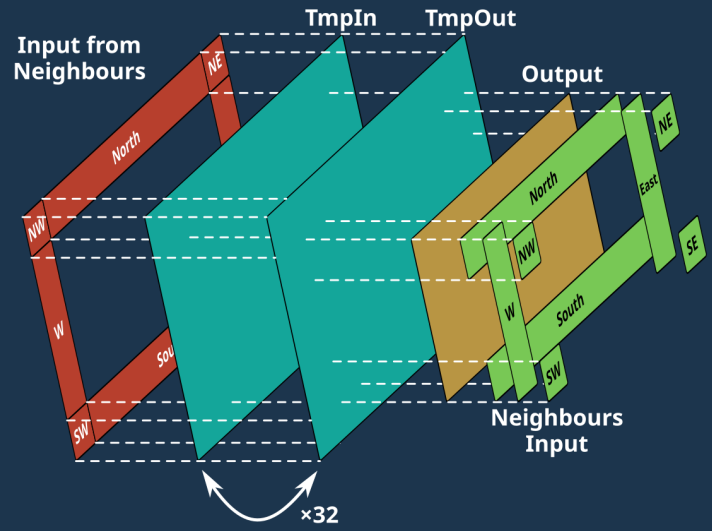
Pyramid Data

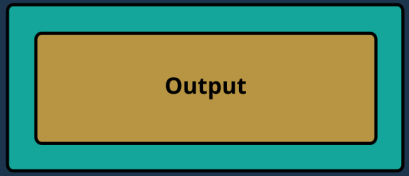


Pyramid Data

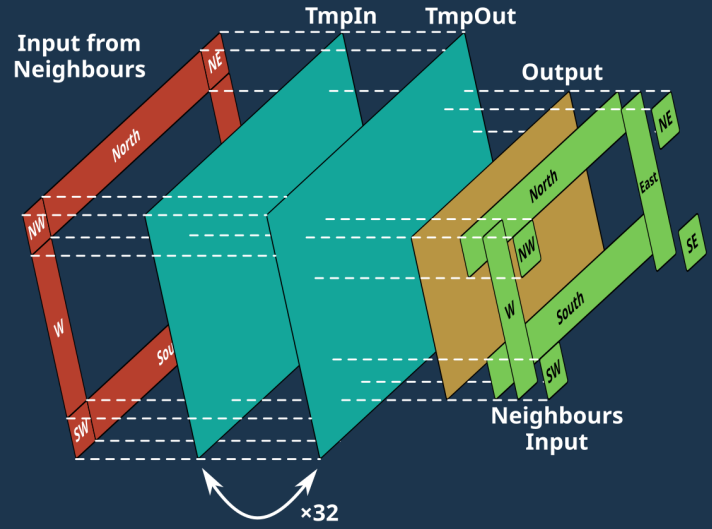


Pyramid Data

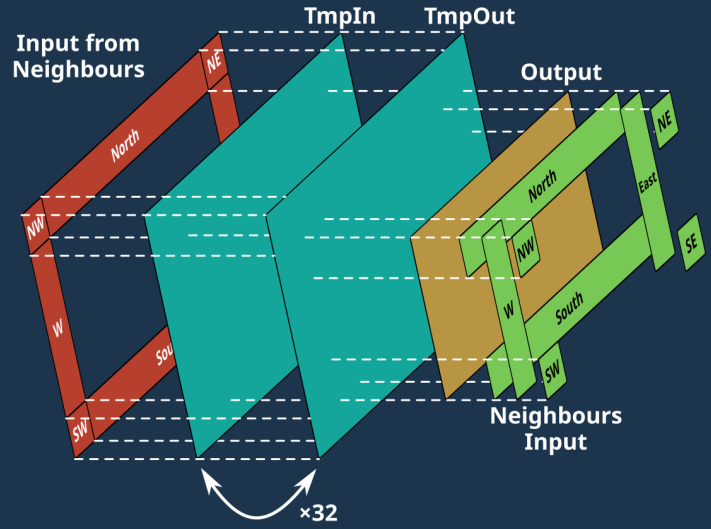




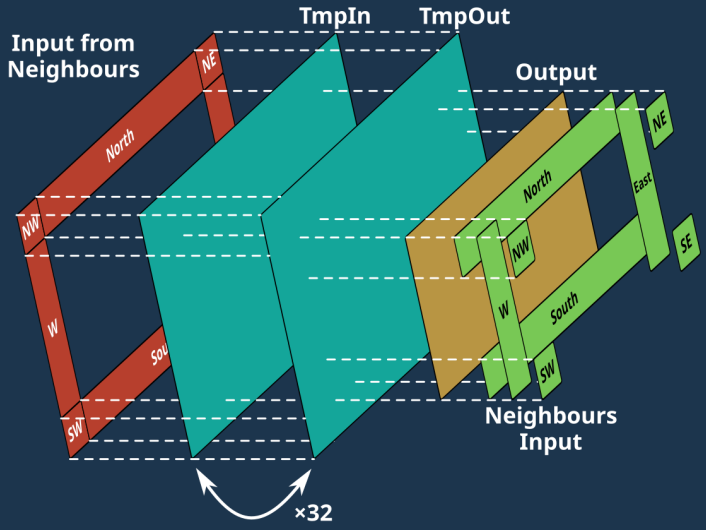
Pyramid Data

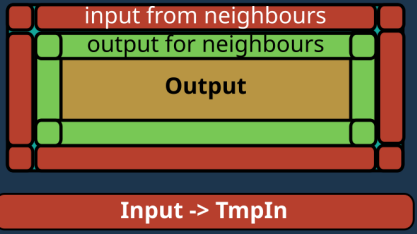


Pyramid Data

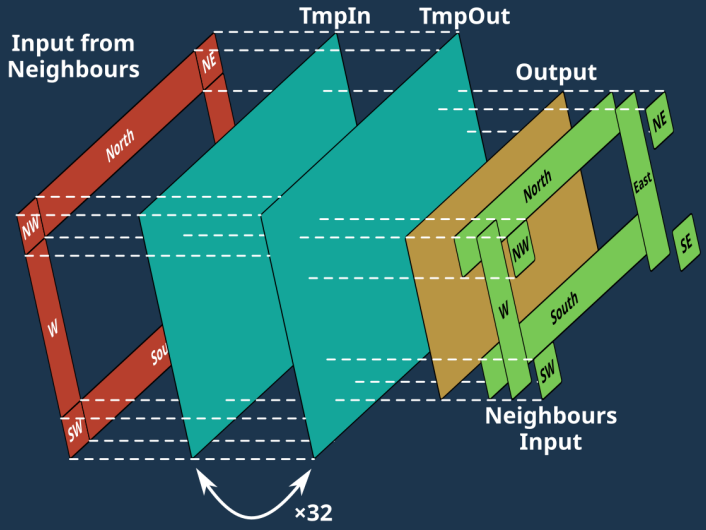


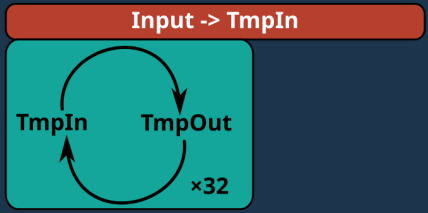
Pyramid Data



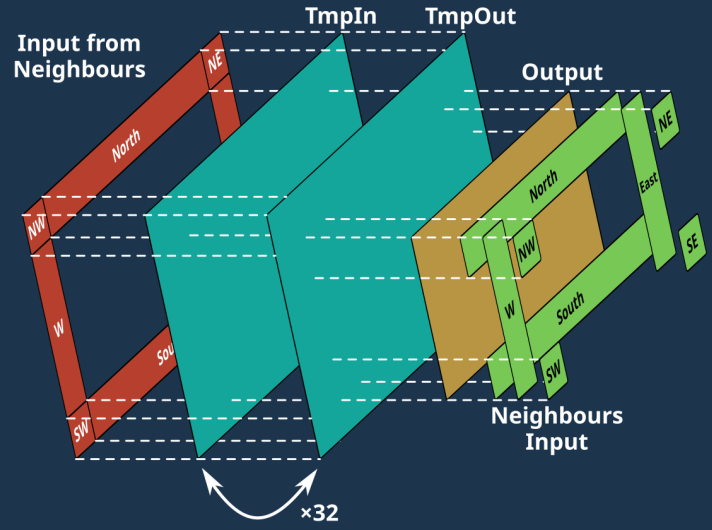


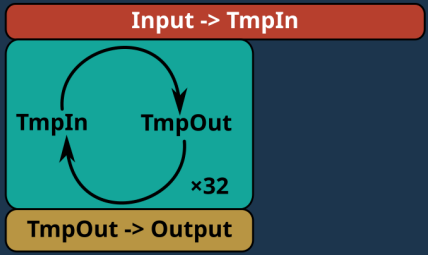
Pyramid Data



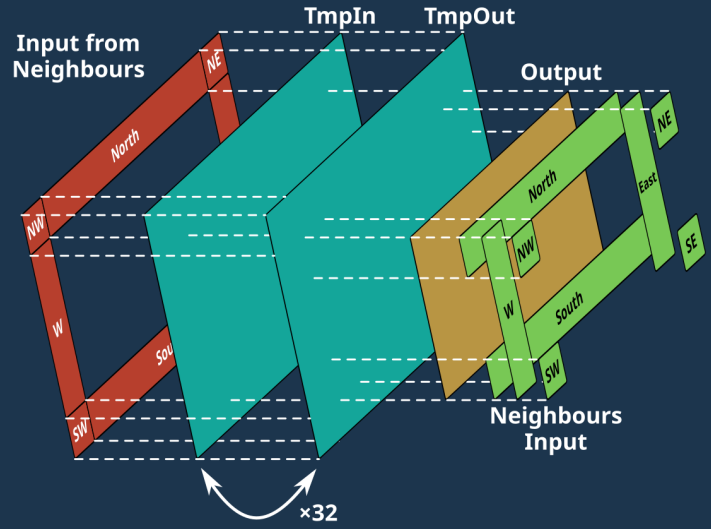


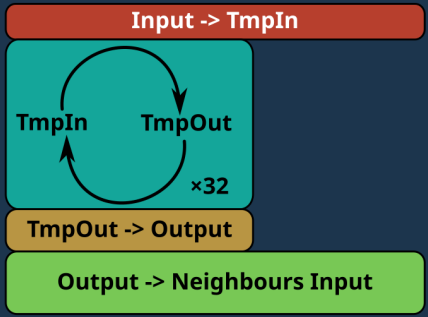
Pyramid Data



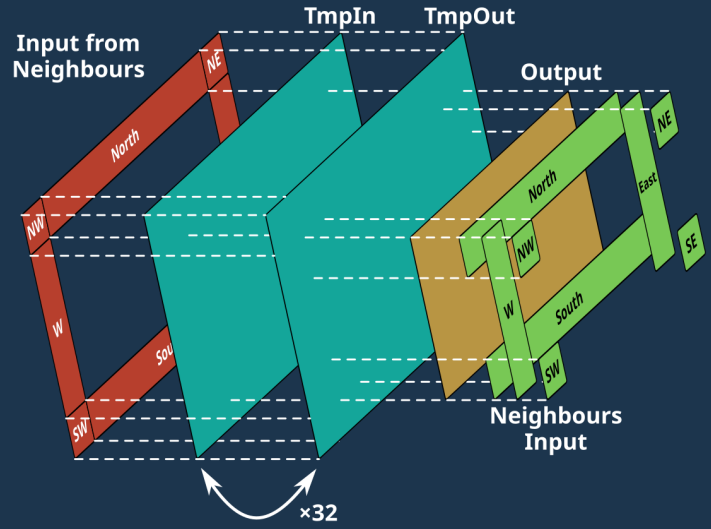


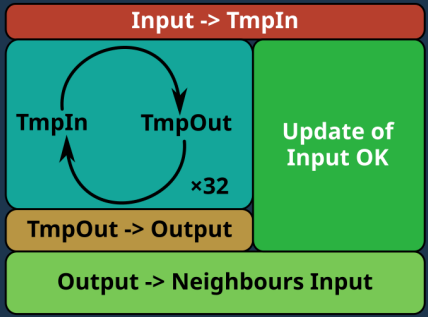
Pyramid Data





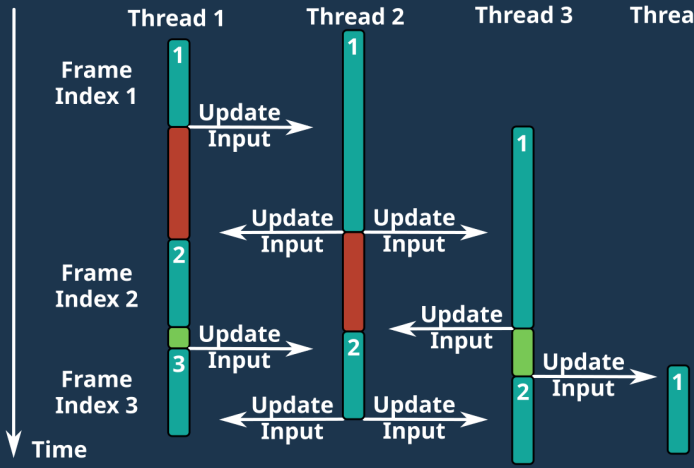
Pyramid Data

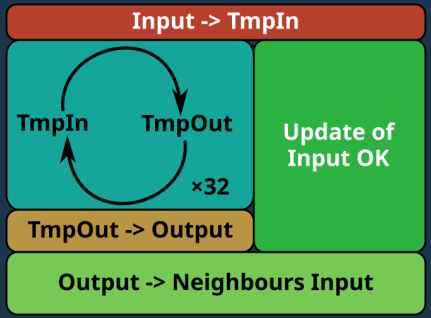




Data Transfer 1d

Computing Wait for input Wait for previous frame





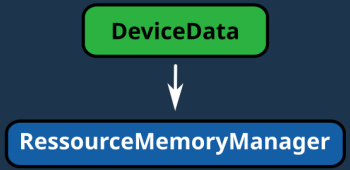
Implementation

Implementation

DeviceData

Tensor of **data** on any kind of device

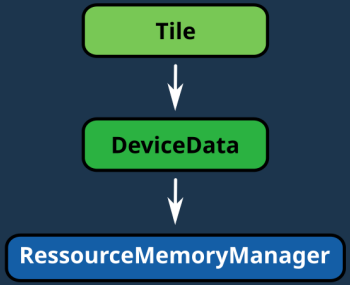
Implementation



Tensor of **data** on any kind of device

Backend for data management **CPU / GPU**
(Intel, AMD, CUDA, HIP, etc)

Implementation

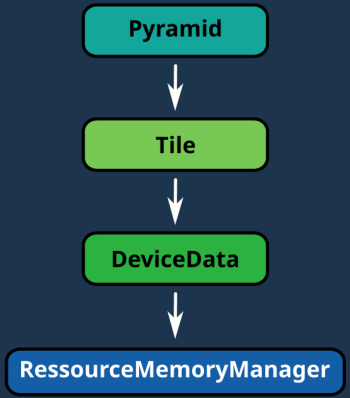


Data aware of its **copy destination** (**push**)

Tensor of **data** on any kind of device

Backend for data management **CPU / GPU**
(Intel, AMD, CUDA, HIP, etc)

Implementation



Description of **computing block**

Data aware of its **copy destination** (**push**)

Tensor of **data** on any kind of device

Backend for data management **CPU / GPU**
(Intel, AMD, CUDA, HIP, etc)

Implementation

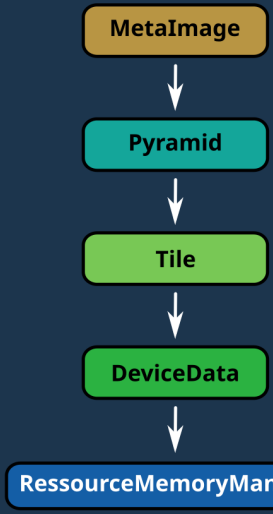


Image to be computed :
- **CPU**
- **GPU**

Description of **computing block**

Data aware of its **copy destination** (**push**)

Tensor of **data** on any kind of device

Backend for data management **CPU / GPU** (Intel, AMD, CUDA, HIP, etc)

Conclusion

Theorically quite efficient => **Multiple GPU streams**

Still need to test with **multiple-GPU**

Also with **AMD HIP**