Supervisors :

- M. CHOUH Hamza
- M. BERGEAUD Vincent

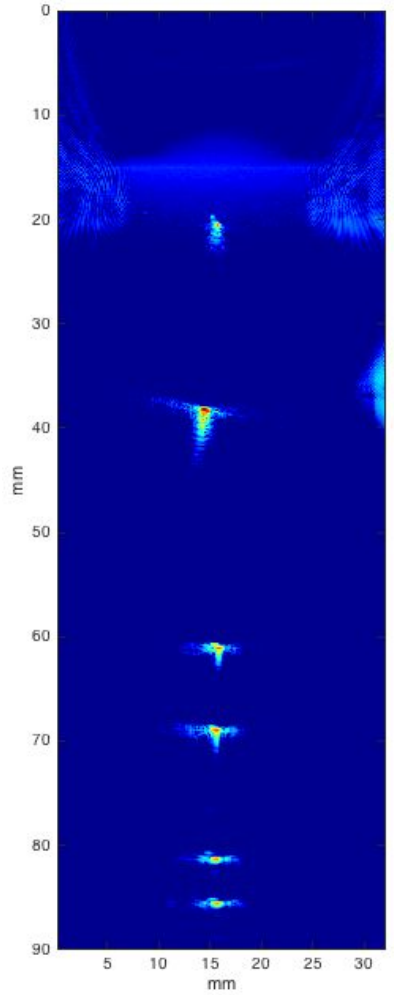
DE LA RECHERCHE À L'INDUSTRIE

Realize an adaptive sampling algorithm for TFM imaging on GPU

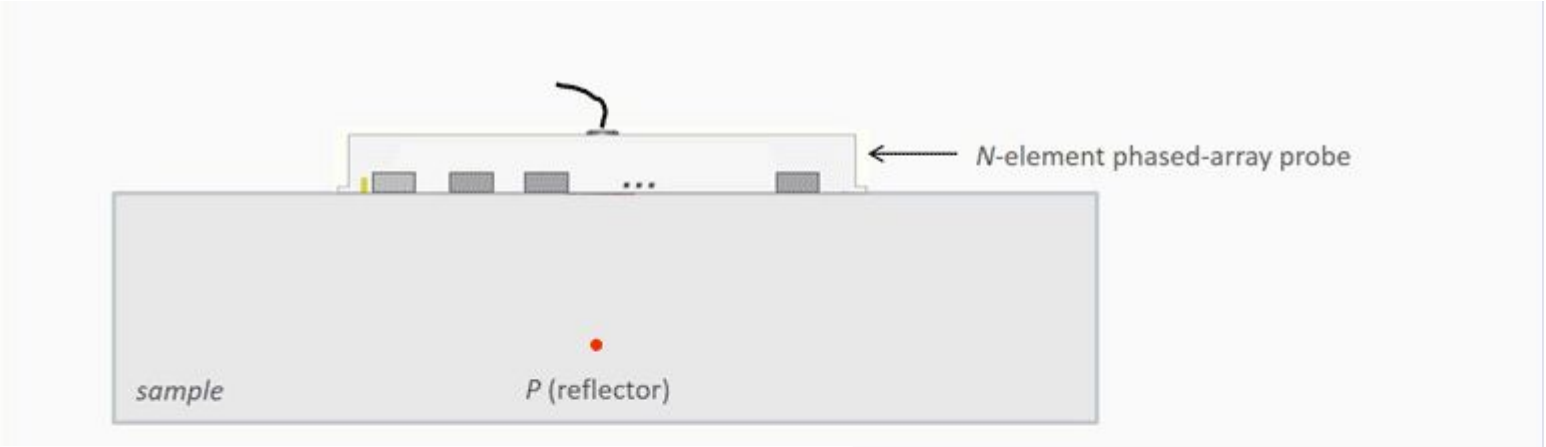
05 juillet 2019

DAVID Jean-François

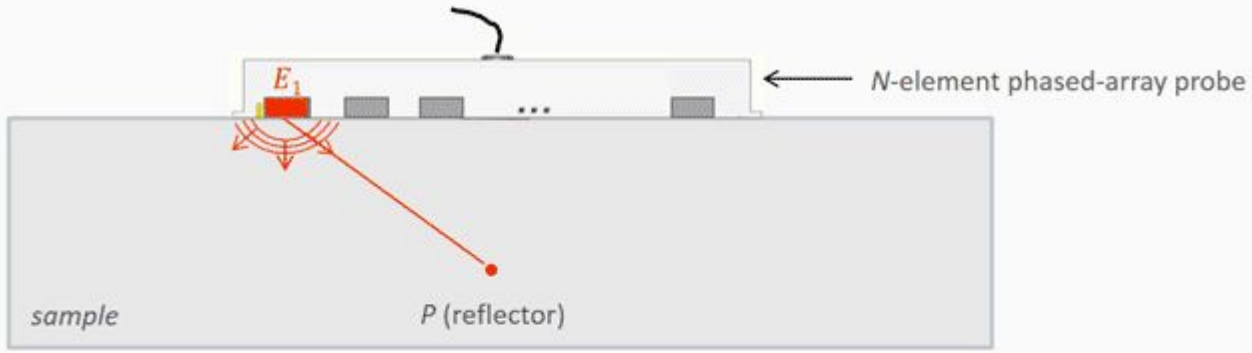
Non-destructive testing by ultrasound



The FMC aquisition

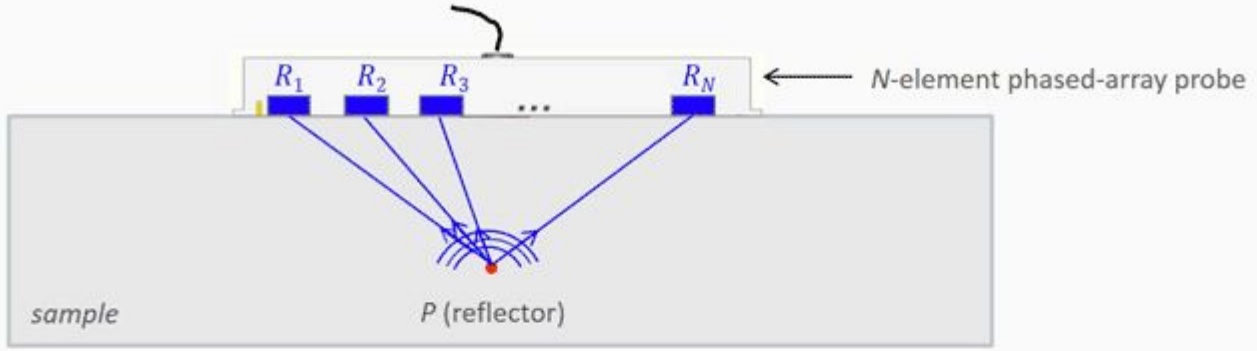


The FMC aquisition



› Element n°1 fires

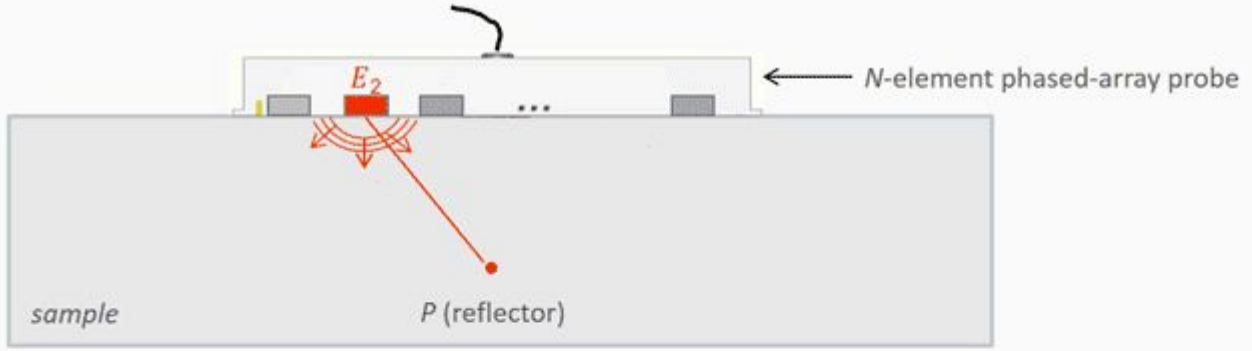
The FMC aquisition



Element n°1 fires

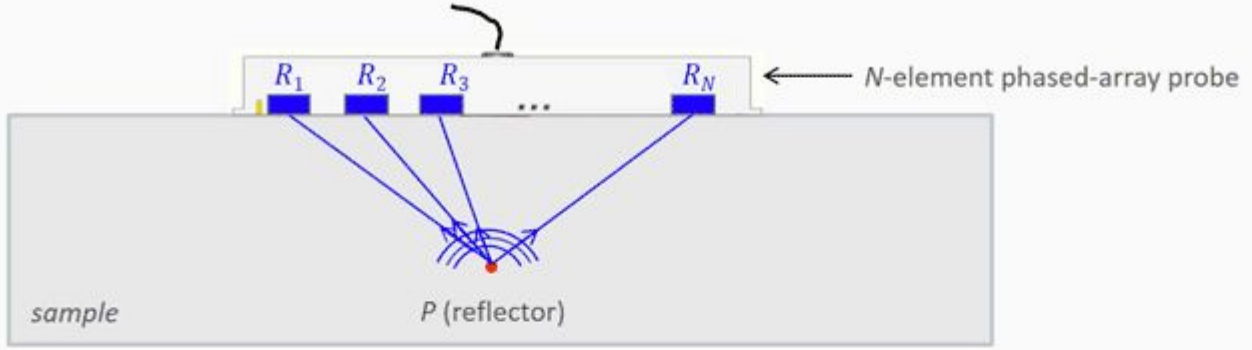
N elements receive, we obtain a first Bscan

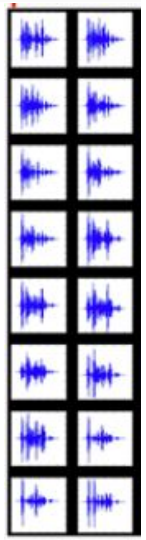
The FMC aquisition



- Element n°1 fires
- N elements receive, we obtain a first Bscan
- Element n°2 fires

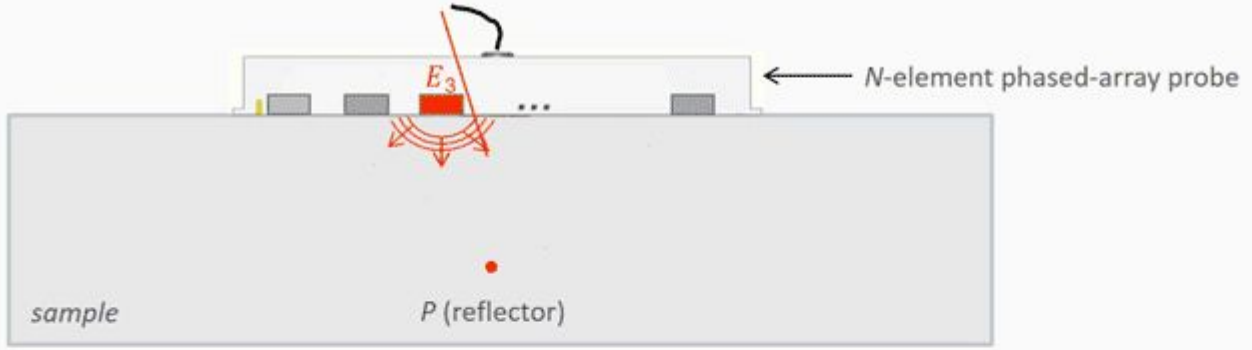
The FMC aquisition

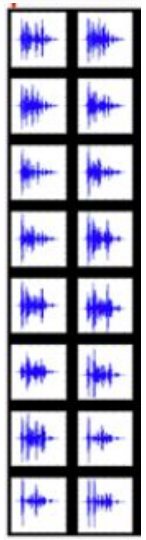




- Element n°1 fires
- N elements receive, we obtain a first Bscan
- Element n°2 fires
- N elements receive, we obtain a 2nd Bscan

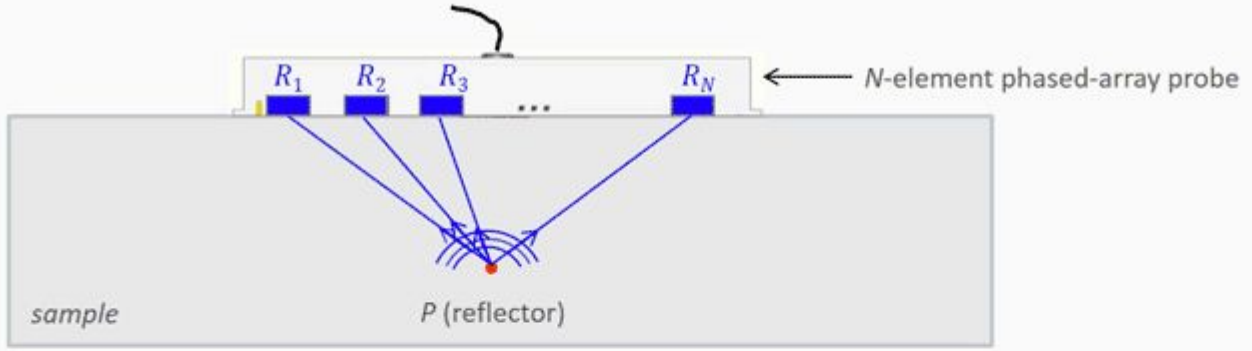
The FMC aquisition

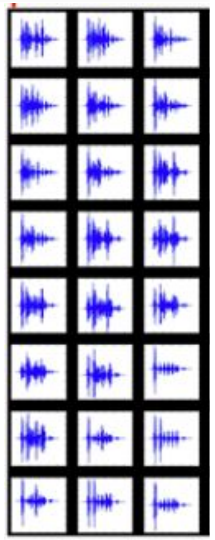




- Element n°1 fires
- N elements receive, we obtain a first Bscan
- Element n°2 fires
- N elements receive, we obtain a 2nd Bscan
- Element n°3 fires

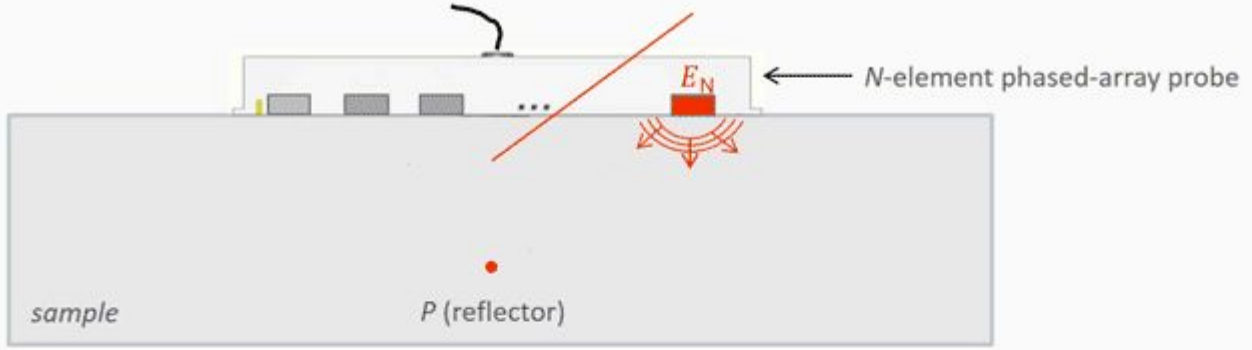
The FMC aquisition

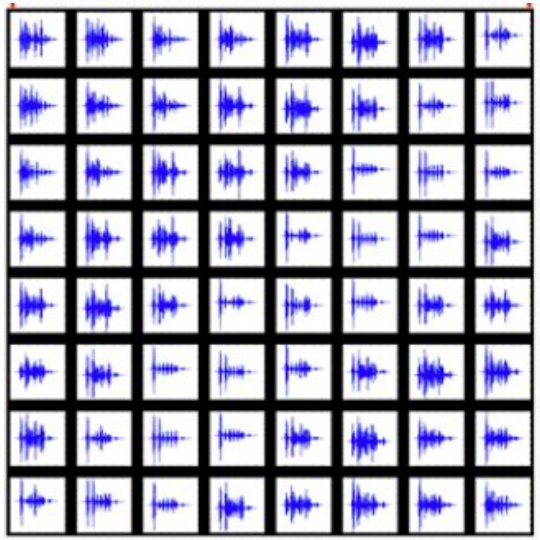




- Element n°1 fires
- N elements receive, we obtain a first Bscan
- Element n°2 fires
- N elements receive, we obtain a 2nd Bscan
- Element n°3 fires
- N elements receive, we obtain a 3rd Bscan

The FMC aquisition



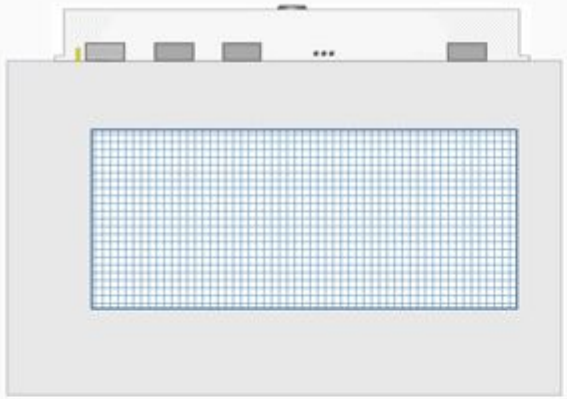


- Element n°1 fires
- N elements receive, we obtain a first Bscan
- Element n°2 fires
- N elements receive, we obtain a 2nd Bscan
- Element n°3 fires
- N elements receive, we obtain a 3rd Bscan

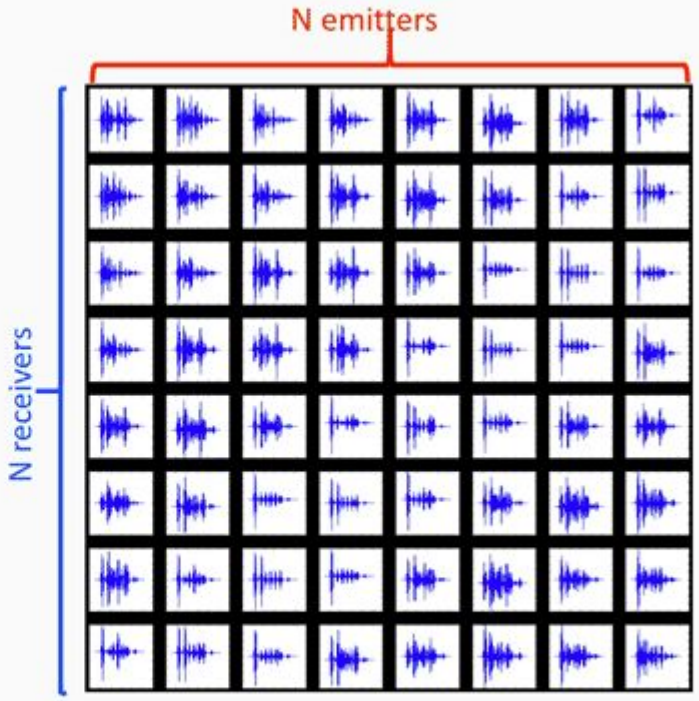
⋮

- Element n°N fires

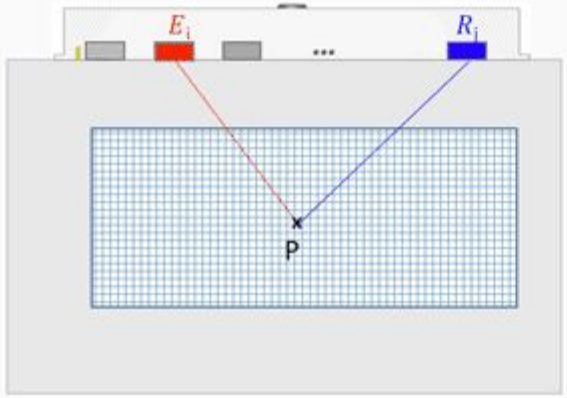
The TFM



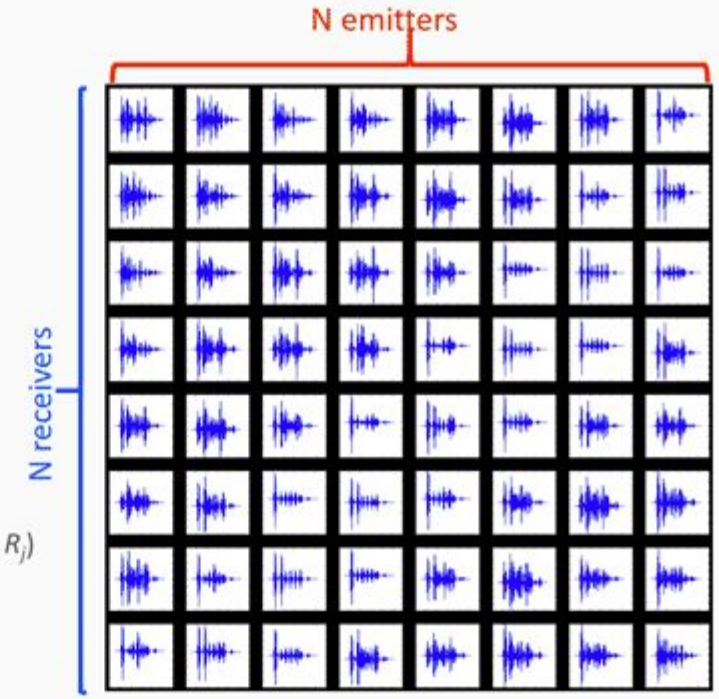
1. Define a region of interest in the component



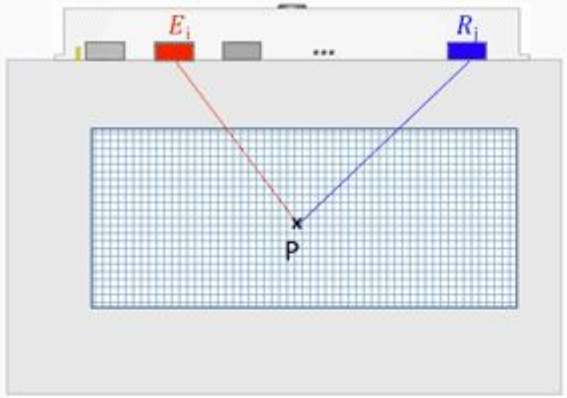
The TFM



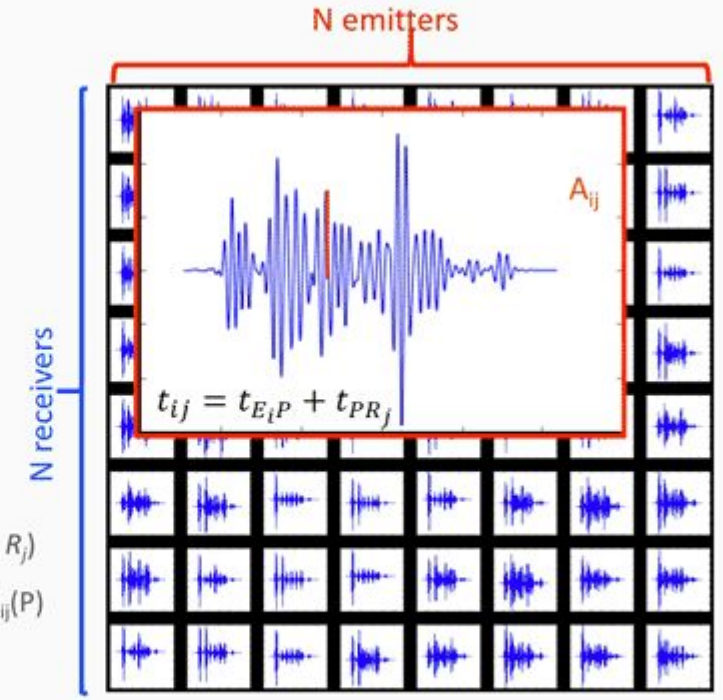
1. Define a region of interest in the component
2. Time-of-flight calculation $t_{ij}(P)$ for all couple (E_i, R_j)



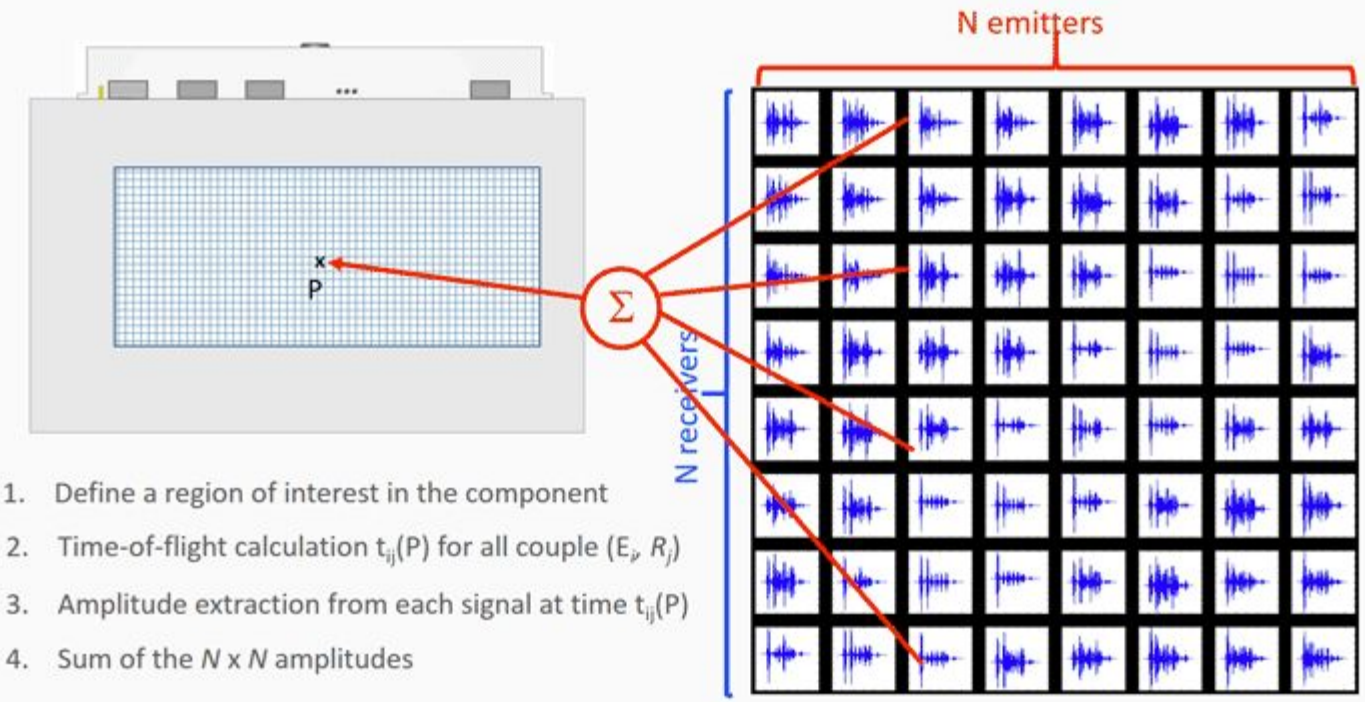
The TFM



1. Define a region of interest in the component
2. Time-of-flight calculation $t_{ij}(P)$ for all couple (E_i, R_j)
3. Amplitude extraction from each signal at time $t_{ij}(P)$

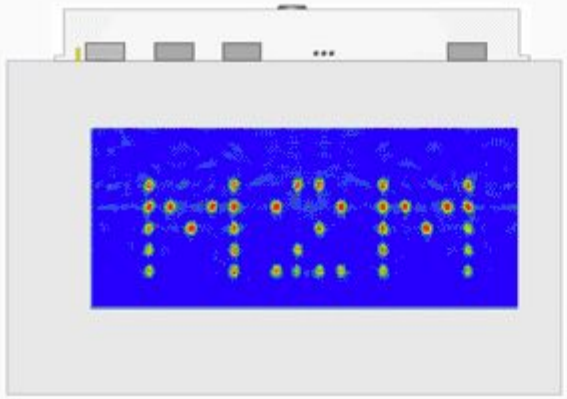


The TFM

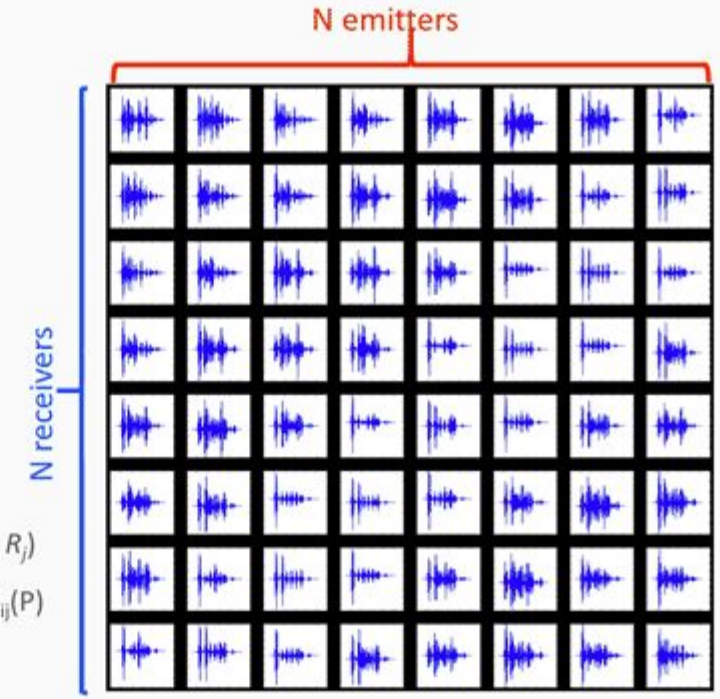


1. Define a region of interest in the component
2. Time-of-flight calculation $t_{ij}(P)$ for all couple (E_i, R_j)
3. Amplitude extraction from each signal at time $t_{ij}(P)$
4. Sum of the N x N amplitudes

The TFM



1. Define a region of interest in the component
2. Time-of-flight calculation $t_{ij}(P)$ for all couple (E_i, R_j)
3. Amplitude extraction from each signal at time $t_{ij}(P)$
4. Sum of the N x N amplitudes
5. Perform the steps above for all pixels



Context

- Work on the thesis of Mr. CHOUH Hamza
  - Approach on CPU (Quadtree and B-SPlines)
  - Perspectives on GPU

- Computing complexity TFM: **PN^2**
  - P: Number of pixels in the TFM image
  - N: Number of elements

- Computing constraints TFM:
  - Real time
  - On-board devices
  - On graphic processor (GPU)
  - Need accurate defect imaging

CUDA

The constraints :

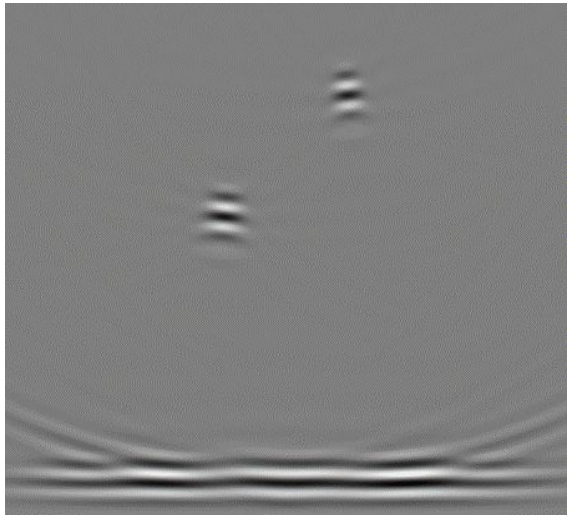
- The SIMD architecture is not very conducive to algorithms with divergent instructions (if/else, while, ect)
- No dynamic allocation, no queue, stack, etc
- Problems of concurrency, writing and reading
- Limited local memory size for threads, which greatly affects computing speed
- Dynamic and recursive programming possible but very slow

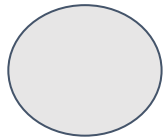
The challenge :

- Develop an adaptive algorithm on GPU
- Find a reconstruction method for non-regular sampling

Quadtree and TFM sampling

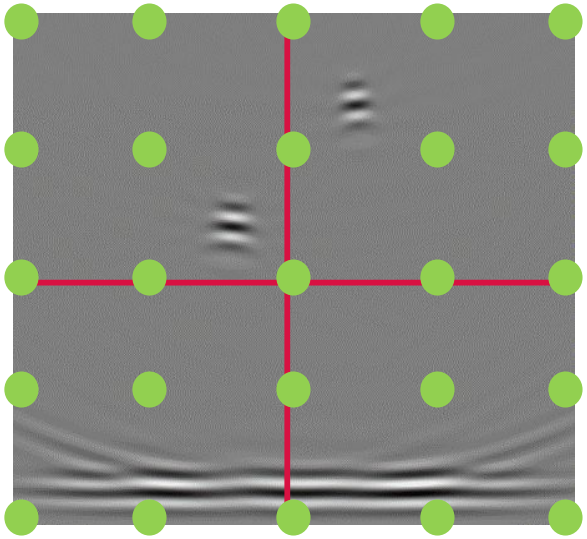
- Is a tree-like data structure in which each node has four children.
- It’s often used to partition a two-dimensional space.

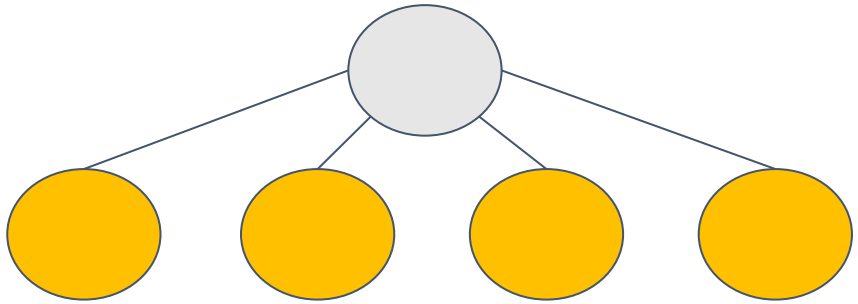




Quadtree and TFM sampling

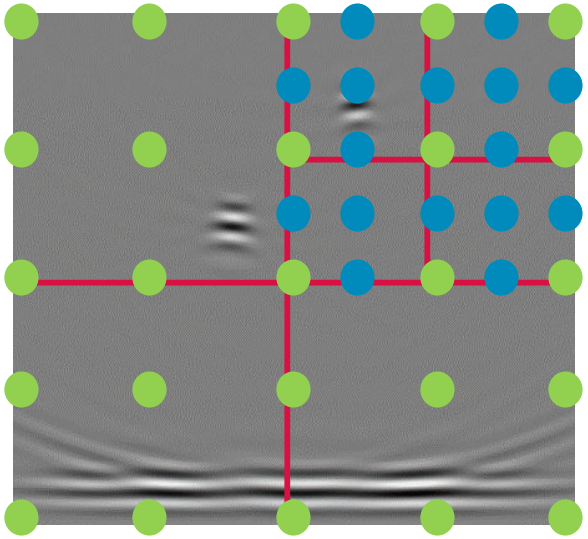
- Is a tree-like data structure in which each node has four children.
- It’s often used to partition a two-dimensional space.

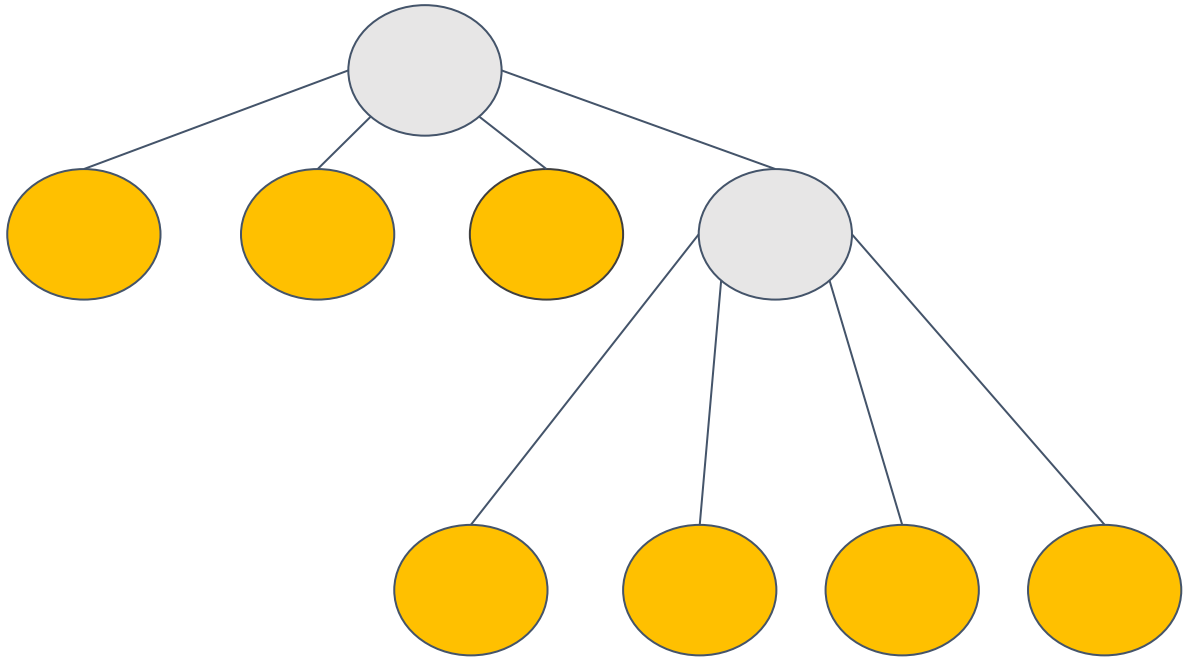




Quadtree and TFM sampling

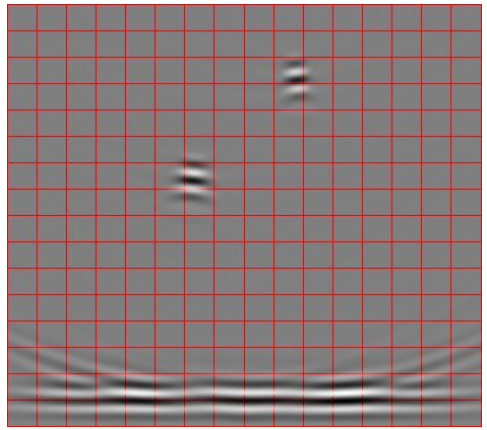
- Is a tree-like data structure in which each node has four children.
- It’s often used to partition a two-dimensional space.





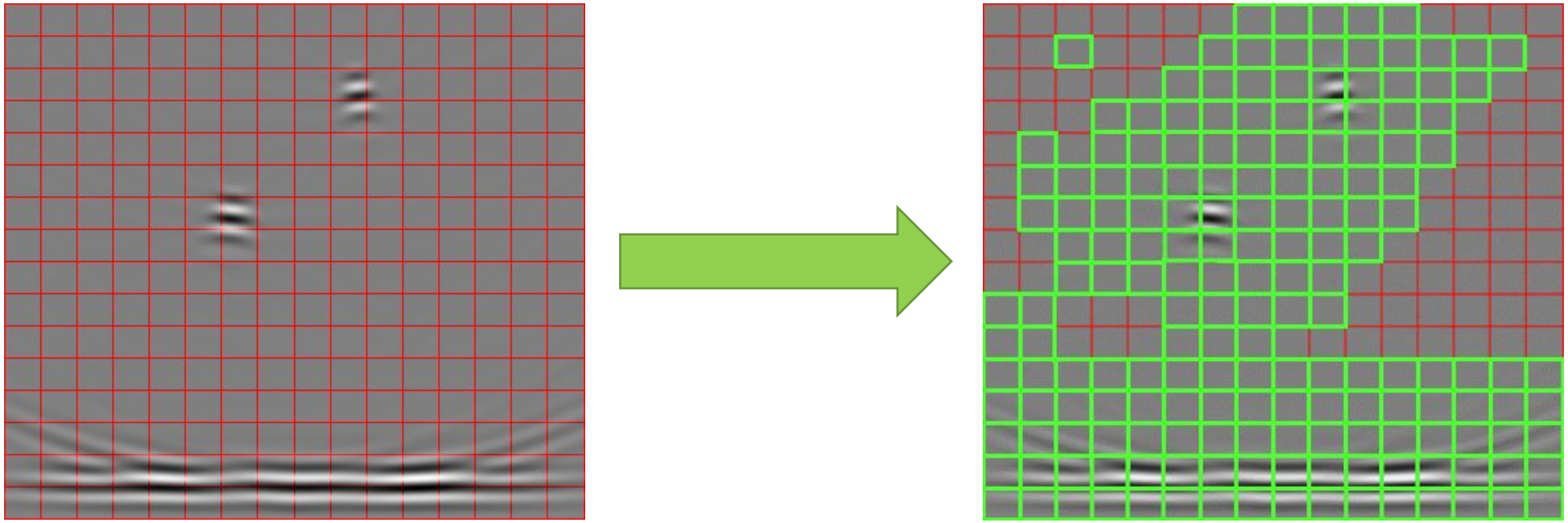
Quadtree initialization

- The array containing the future nodes is pre-allocated, as well as an array of booleans allowing to know the TFM points that have been calculated.
- The root node is initialized, then it is subdivided, as well as its children, until it reaches a certain depth to load the GPU as much as possible.

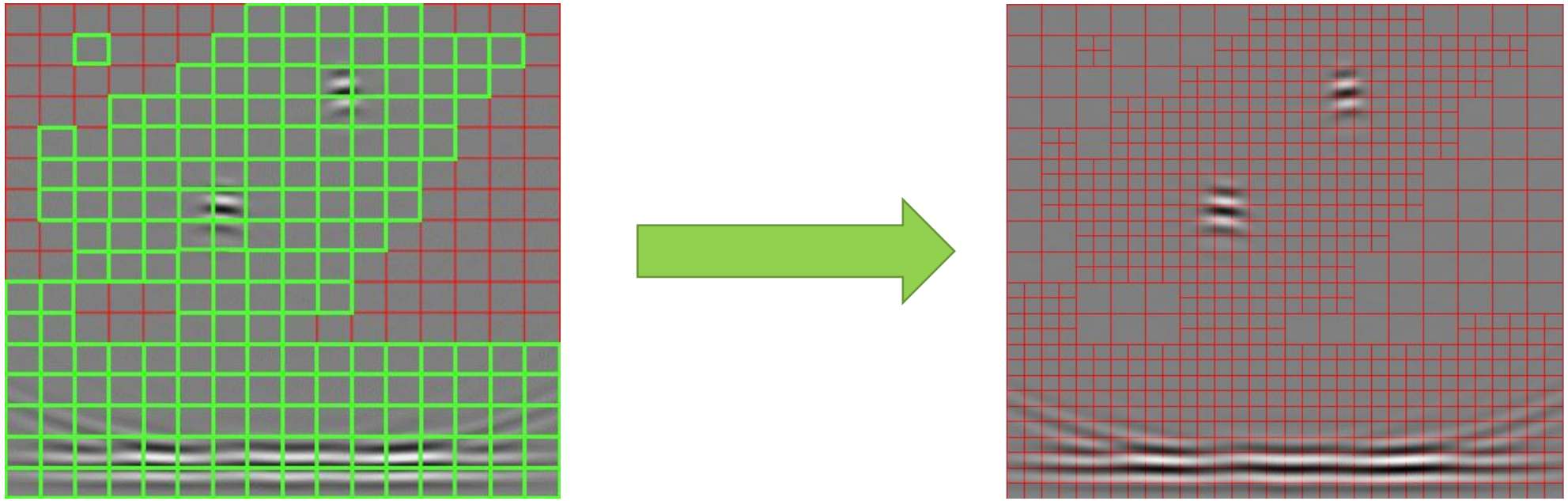


Selection of areas to be calculated

- The score of each node is calculated according to its best amplitude and depth in the quadtree.
- The nodes are sorted by their score in a table.
- A fixed number of nodes with the highest score are selected from the table. These are the nodes that will be subdivided.

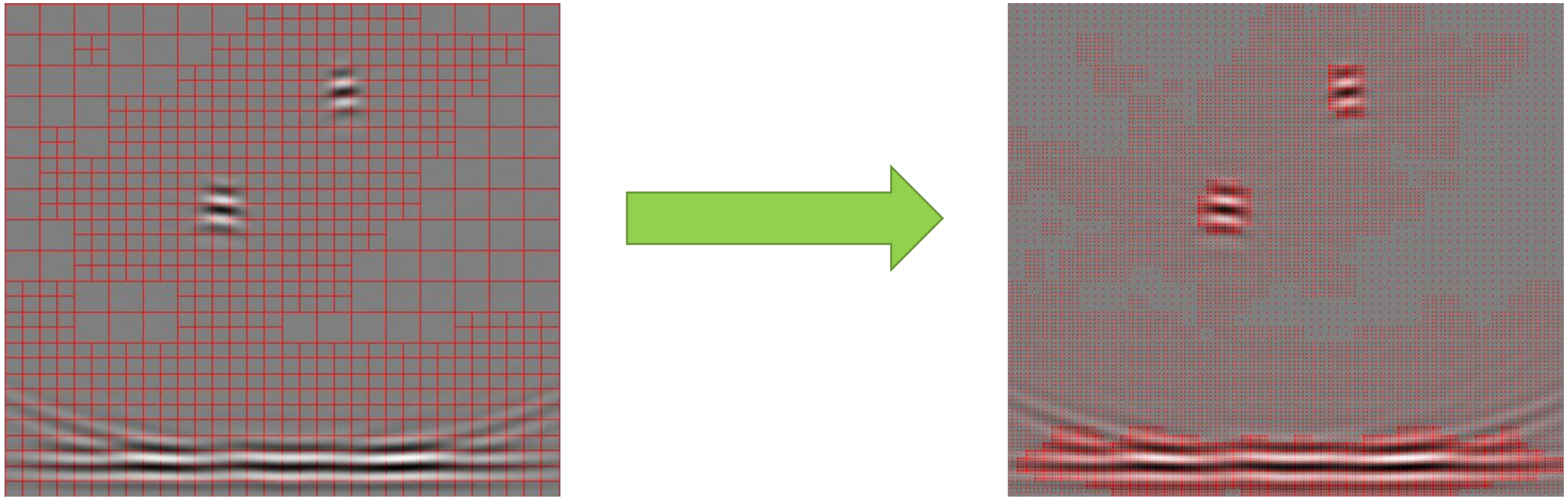


Subdivision of nodes in these areas



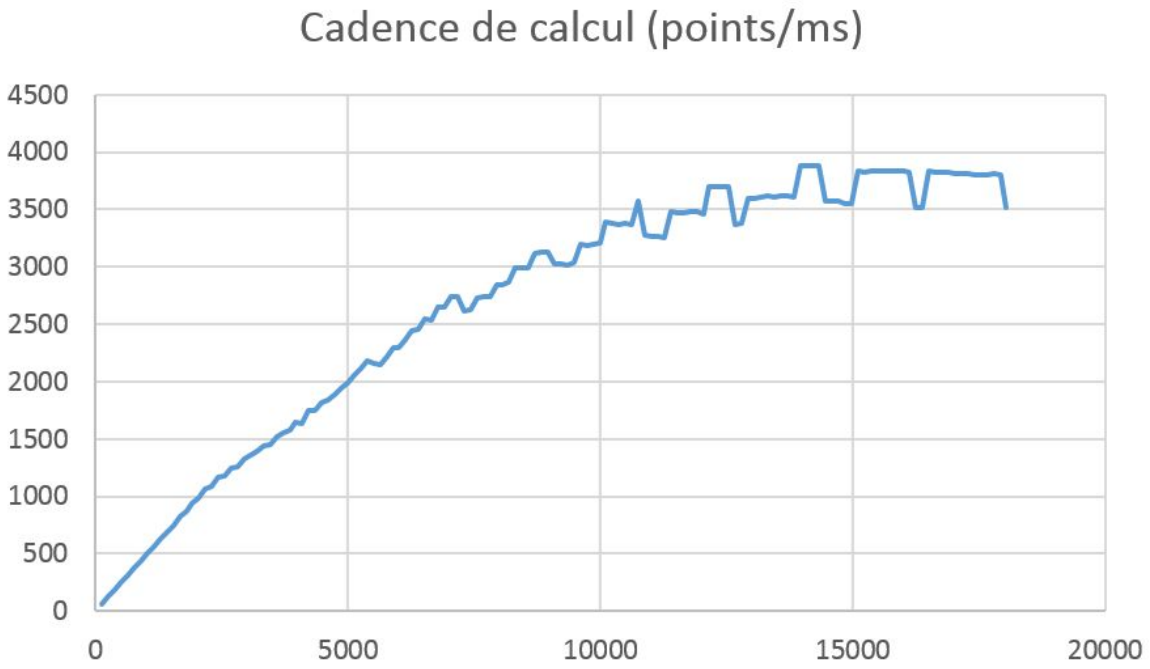
Calculation of the TFM

- The amplitude value of the TFM is calculated for the pixels corresponding to each of the cardinal points of the nodes.



5. Le critère d'arrêt

- The user sets a target frame rate (e.g. 50 fps)
- We proceed to a first estimation of the calculation rate which allows us to determine the "budget" of points to calculate per image
  - Measurement of the computation time for a group of pixels at each iteration
  - We give a maximum time for the calculation of an image
- We look for the lowest possible granularity value that maximizes the rate.
  - It must load the GPU
  - Must be located on a peak rate

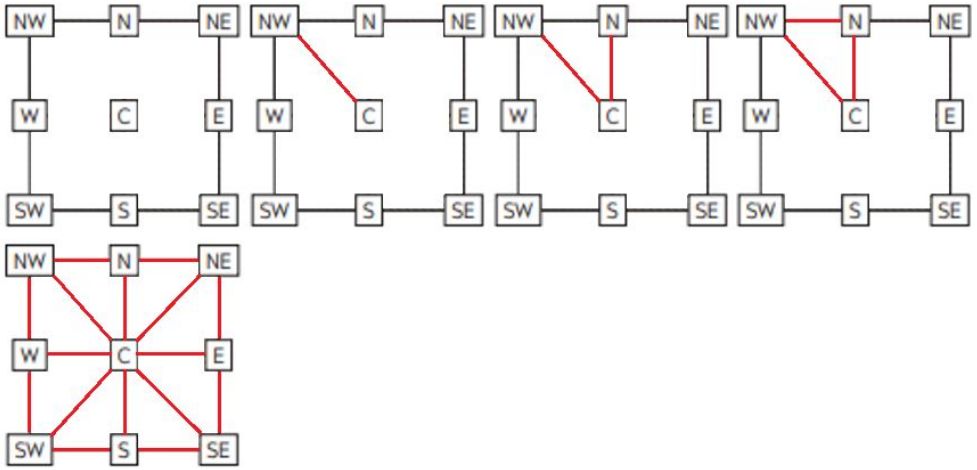


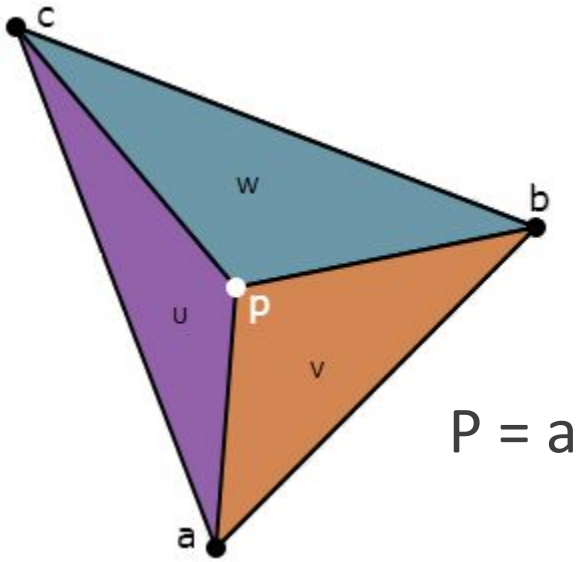
Reconstruction of an incomplete image

- The reconstruction of the image is done in two phases: triangulation (mesh of the irregular grid of pixels) and interpolation

- The triangulation is done on each leaf node of the quadtree, each vertex of the triangles corresponds to a TFM point calculated beforehand

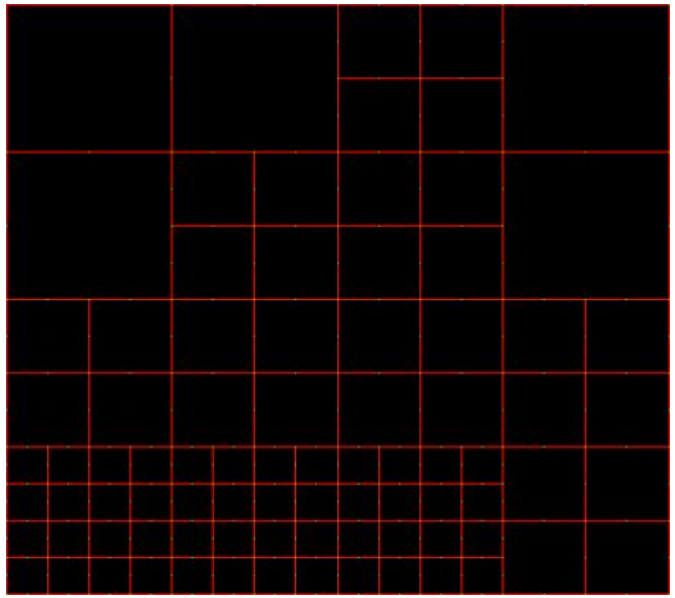
- The interpolation used is a barycentric interpolation

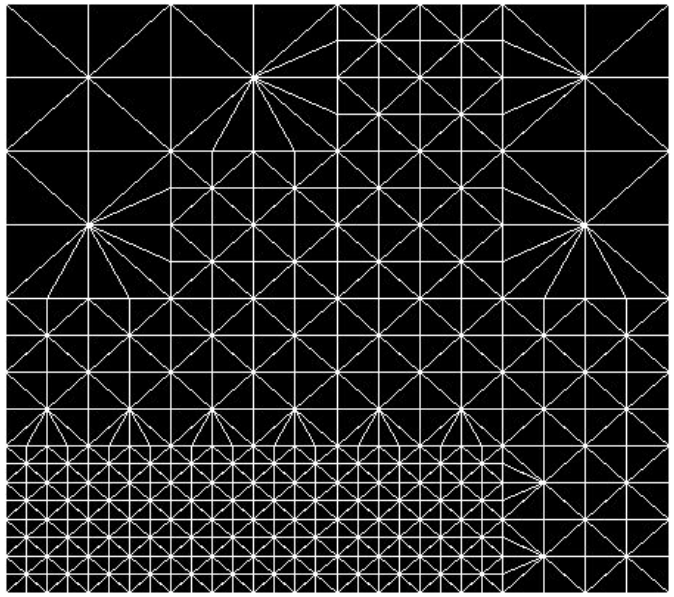


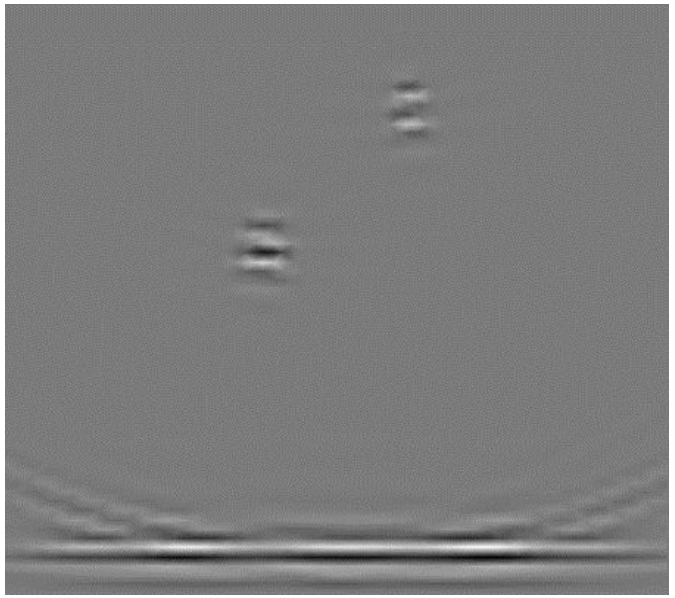


$$P = aw + bu + cv$$

Reconstruction of an incomplete image



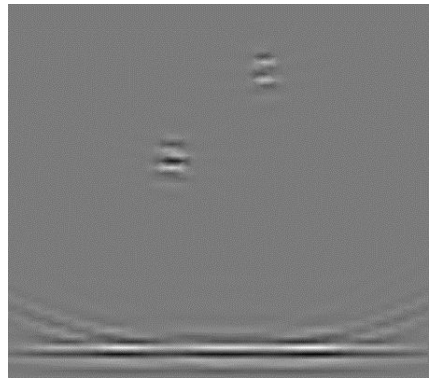


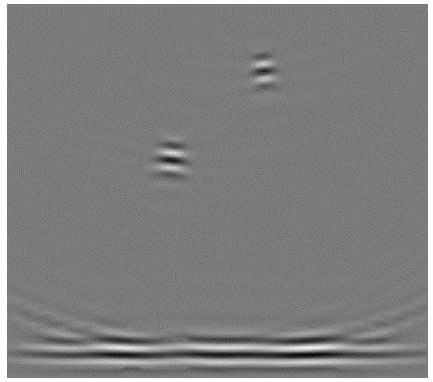


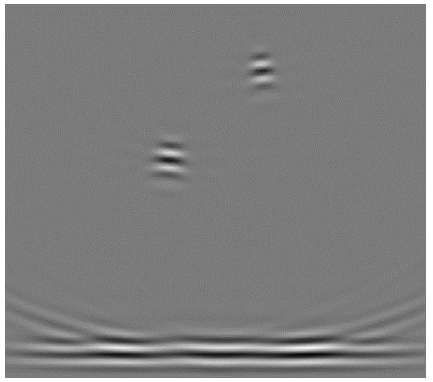
Adaptive refinement

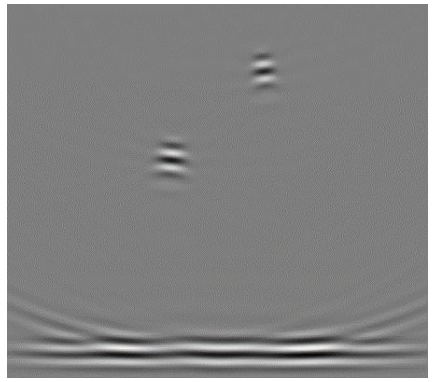
image size 451 x 401 = 180 851 pixels

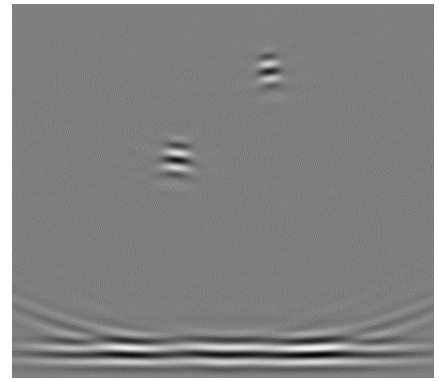
3 002 samples 8 ms	5 968 samples 10 ms	13 393 samples 18 ms	19 314 samples 22 ms	Idéale 70 ms (without spl)

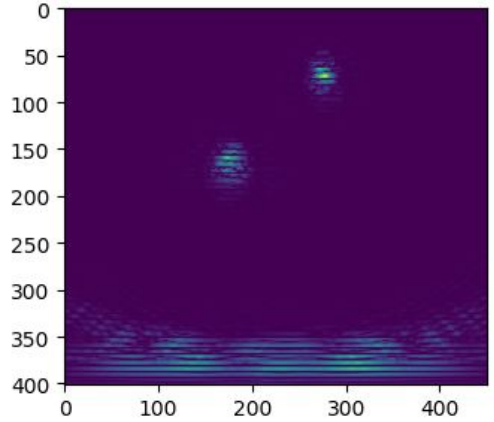




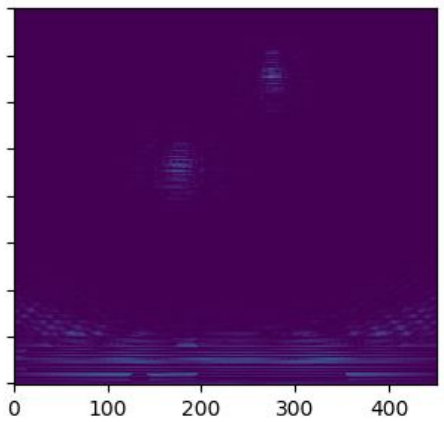




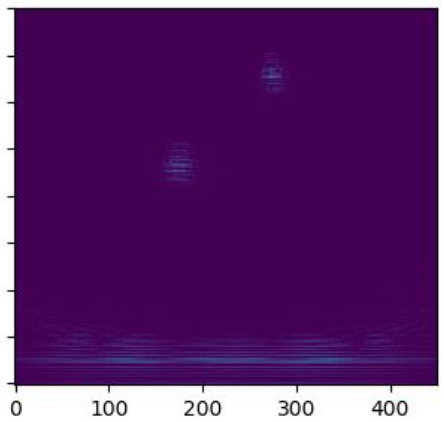




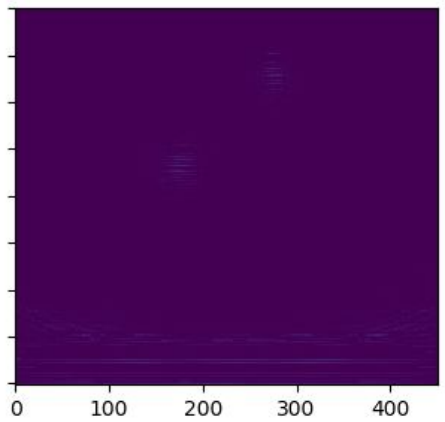
Error max : 65%



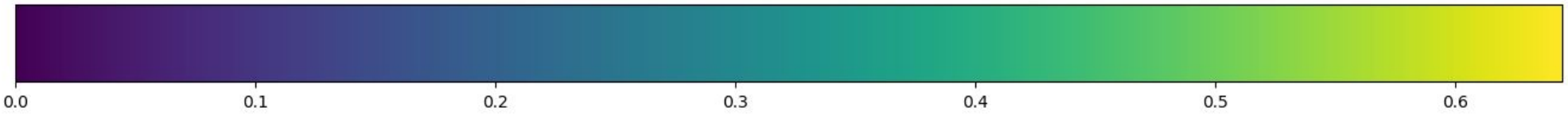
Error max : 46%



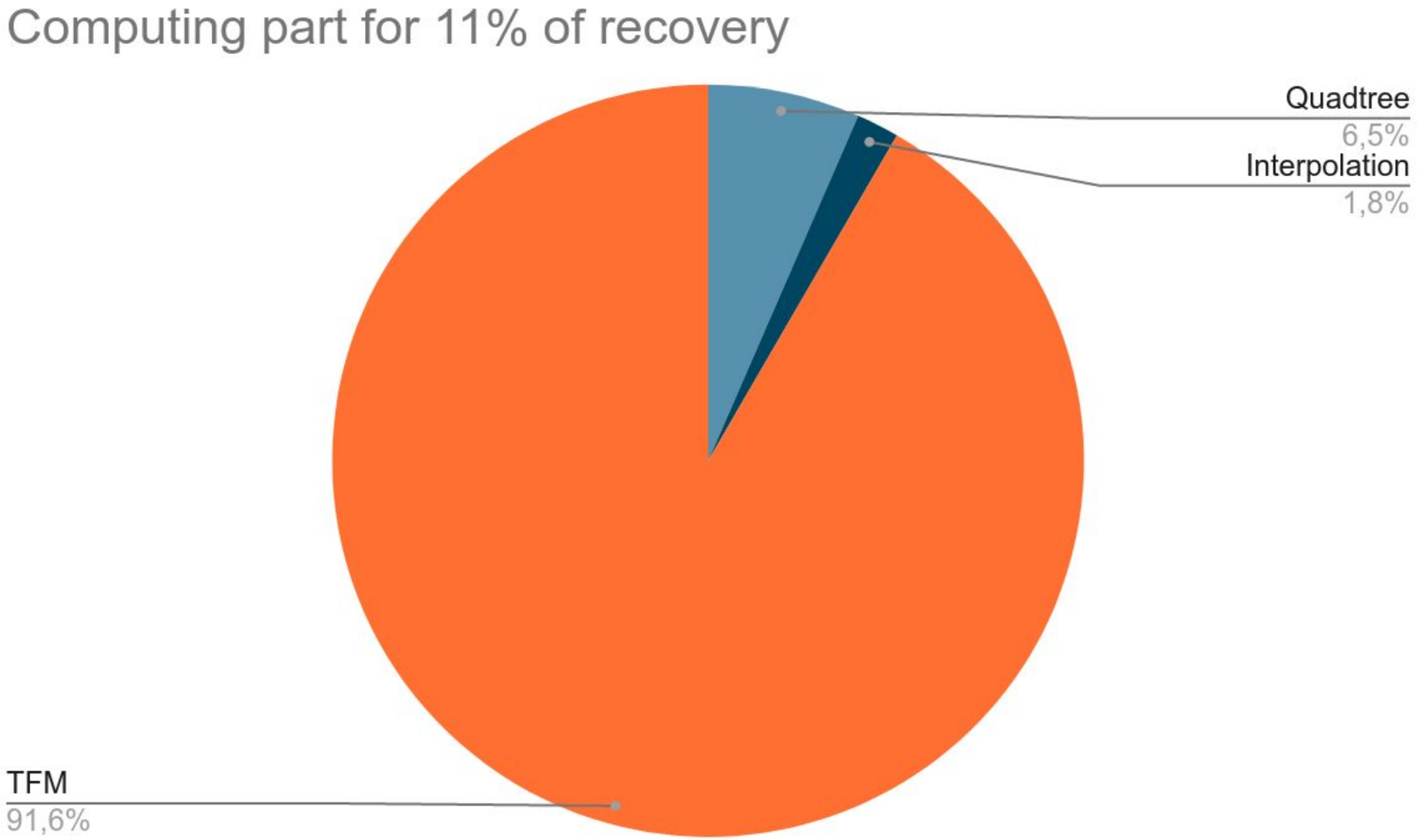
Error max : 24%



Error max : 8%

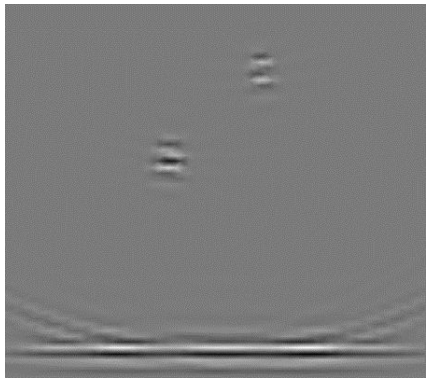
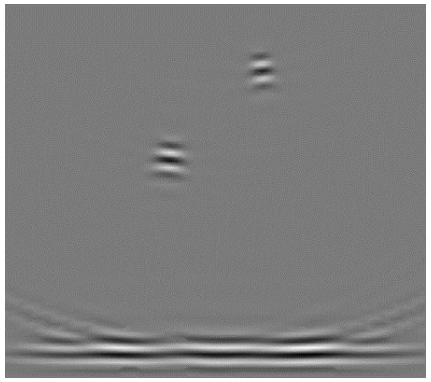
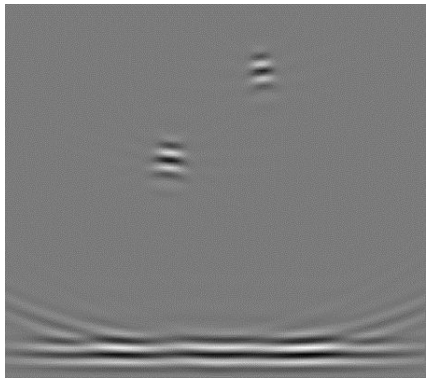
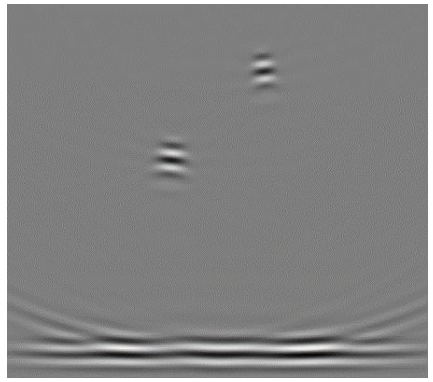
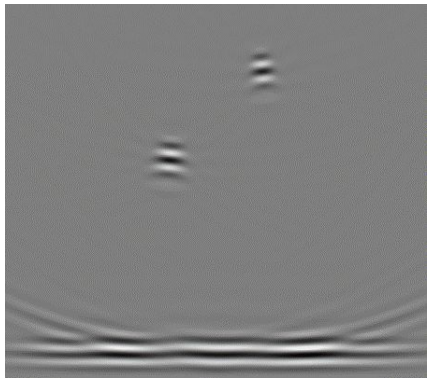
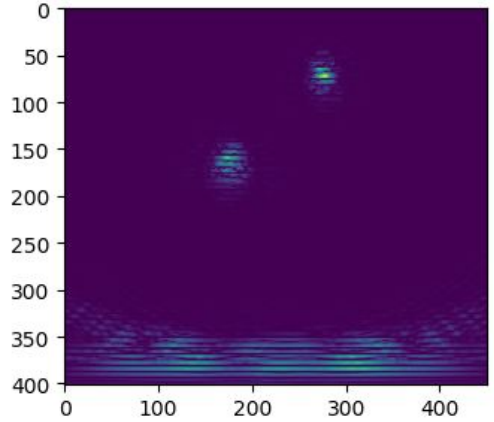
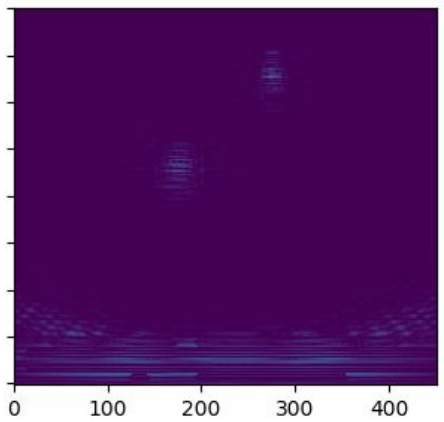
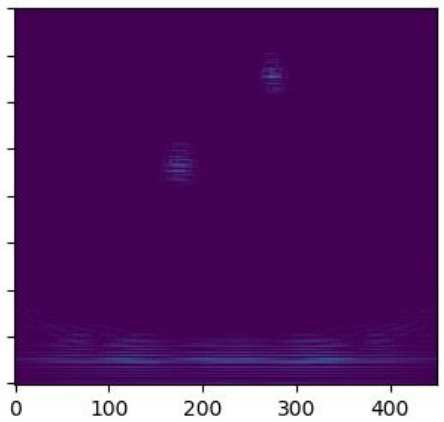
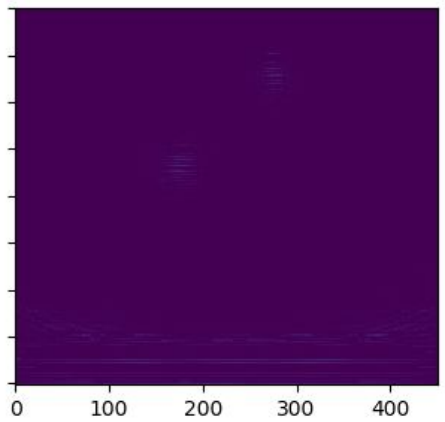


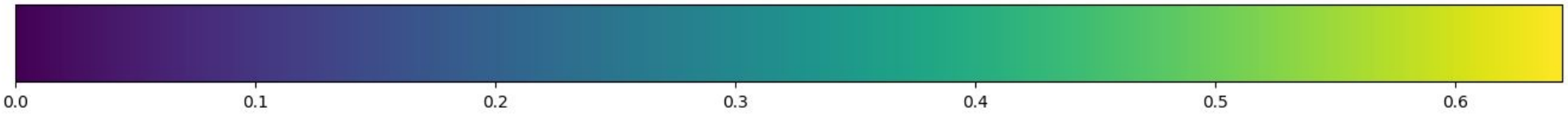
Result


Computing part for 11% of recovery
Quadtree
6,5%
Interpolation
1,8%
TFM
91,6%

Raffinement adaptatif

image de taille 451 x 401 = 180 851 pixels

3 002 samples 10 ms	5 968 samples 14 ms	13 393 samples 18 ms	19 314 samples 22 ms	Idéale 70 ms
				
				
Error max : 65%	Error max : 46%	Error max : 24%	Error max : 8%	



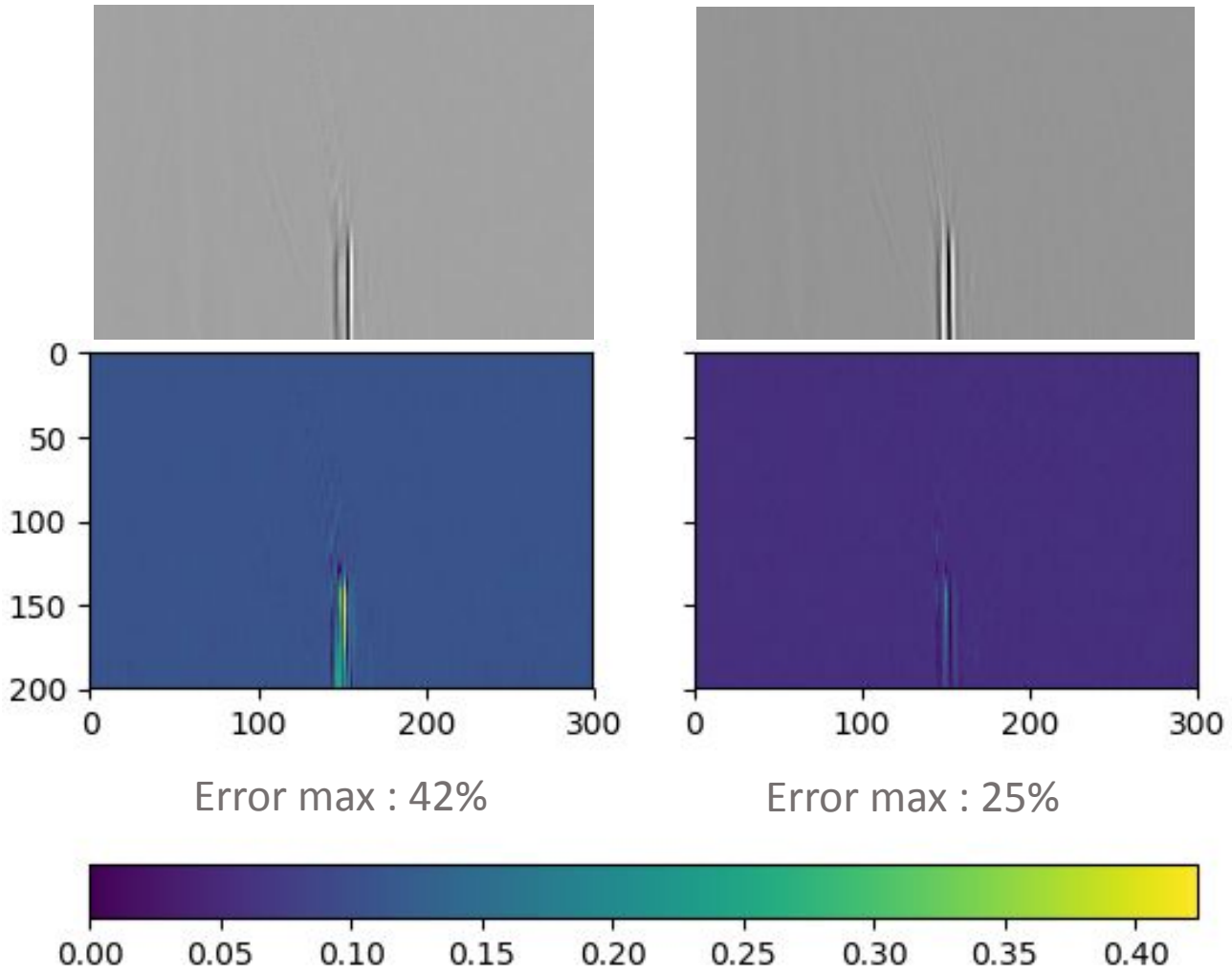
Other exemple

image size 301 x 201 = 60 501 pixels

13 049 samples

20 367 samples

Idéale

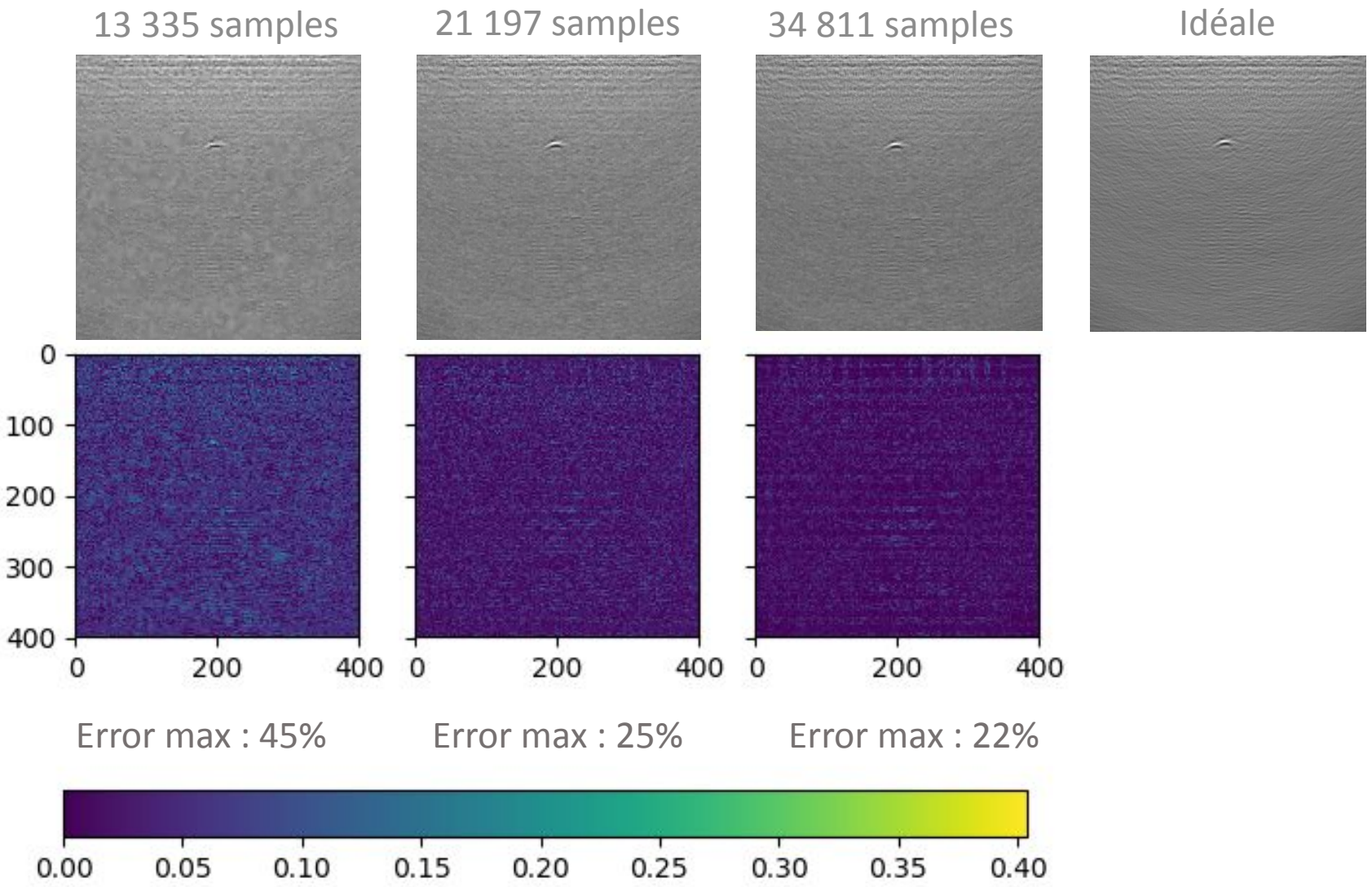


Error max : 42%

Error max : 25%

Other exemple

image size 401 x 401 = 160 801 pixels



Conclusion

- Implementation of an adaptive sampling algorithm, and TFM imaging reconstruction on GPU
- For a small number of samples, correct images are reconstructed
- The construction of the tree and its refinement represent a negligible additional cost compared to TFM

The TFM

```
1 Pour chaque element en emission iE
2     Pour chaque element en reception iR
3         Pour chaque pixel de l'image P
4             Ajouter la contribution du tir (iE,iR) sur le pixel P
```