

# Master 1 – IMAGINE

Semestre 2

## Présentation finale du projet image

Compression basée superpixels

**Groupe 4.3**

# Résumé

- 1 - Présentation de la segmentation en superpixels
- 2 - Présentation de l'algorithme et compression : SLIC
- 3 - Résultats
- 4 - D'autres algorithmes et nos tentatives
- 5 - D'autres possibilités d'utilisations
- 6 - Présentation de l'interface graphique et démonstration

# 1 – Segmentation en superpixels



Diminuer la quantité d'information à traiter en regroupant des pixels. Cela revient à créer des groupes de pixels ayant des valeurs similaires.

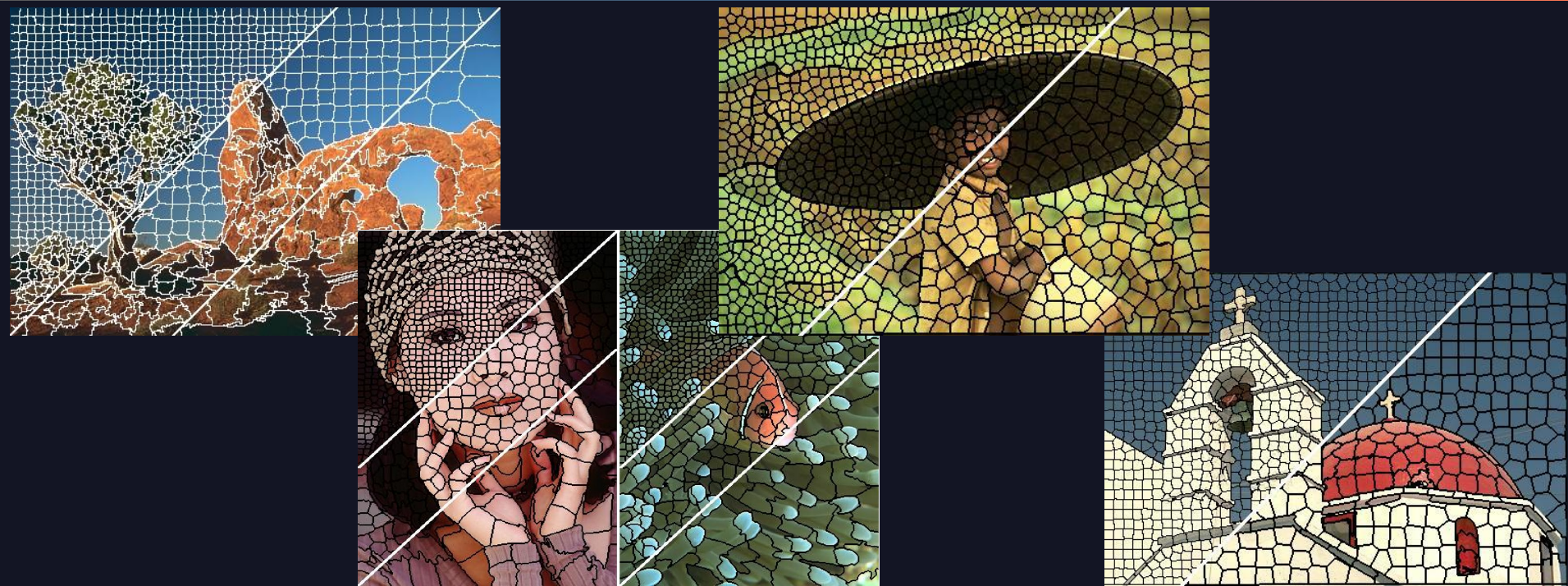
## Permet :

- . D'effectuer de la détection
- . Limiter le traitement à certaines régions clés

## Algorithmes :

- . SLIC / Quickshift / Turbopixels / etc...
- . SNIC / TASP / etc ...

## 2 - SLIC et compression



## 2 – SLIC et compression

### Etapes :

1. Initialisation des centres des clusters (= superpixels)
2. Perturbation des centres des clusters grâce à un gradient
3. Mise à jour des centres des clusters et attribution des couleurs

### On note :

N : Le nombre de pixels d'une image

K : Le nombre de superpixels

Taille de chaque superpixel =  $N / K$

Distance entre chaque superpixel =  $\sqrt{\text{tailleSuperpixel}}$

Distance spatiale :  $d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$

Distance spectrale :  $d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$

Distance totale (5 dimensions : X, Y, R, G, B ou X, Y, L, a, b) :  $D' = \sqrt{\left(\frac{d_s}{m}\right)^2 + \left(\frac{d_{lab}}{S}\right)^2}$

#### Algorithm 1 SLIC superpixel segmentation

```
/* Initialization */
Initialize cluster centers  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  by
sampling pixels at regular grid steps  $S$ .
Move cluster centers to the lowest gradient position in a
 $3 \times 3$  neighborhood.
Set label  $l(i) = -1$  for each pixel  $i$ .
Set distance  $d(i) = \infty$  for each pixel  $i$ .

repeat
/* Assignment */
for each cluster center  $C_k$  do
for each pixel  $i$  in a  $2S \times 2S$  region around  $C_k$  do
Compute the distance  $D$  between  $C_k$  and  $i$ .
if  $D < d(i)$  then
set  $d(i) = D$ 
set  $l(i) = k$ 
end if
end for
end for
/* Update */
Compute new cluster centers.
Compute residual error  $E$ .
```

<https://jakubnowosad.com/giscience-2021/#21>

## 2 – SLIC et compression

### Type de compression :

- . Mise en place d'une compression palette
  - avec génération de la palette de couleurs

**Problème :** Beaucoup de répétitions du même symbole



Mise en place d'un codage par plage  
afin de réduire les symboles

## 2 – SLIC et compression

Métriques :

PSNR (Peak Signal to Noise Ratio) en décibels :

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{255^2}{\text{EQM}} \right)$$

Entropie :

$$- \sum_{i=1}^n P_i \log_b P_i$$

Entropie (RGB) = (Entropie(R) + Entropie(G) + Entropie (B) ) / 3



## 2 – SLIC et compression

### 1. Initialisation des centres des clusters (= superpixels)



**Données :**

$K = 150$

$m = 10$

Voisinage = 3

Nombre d'itérations = 5

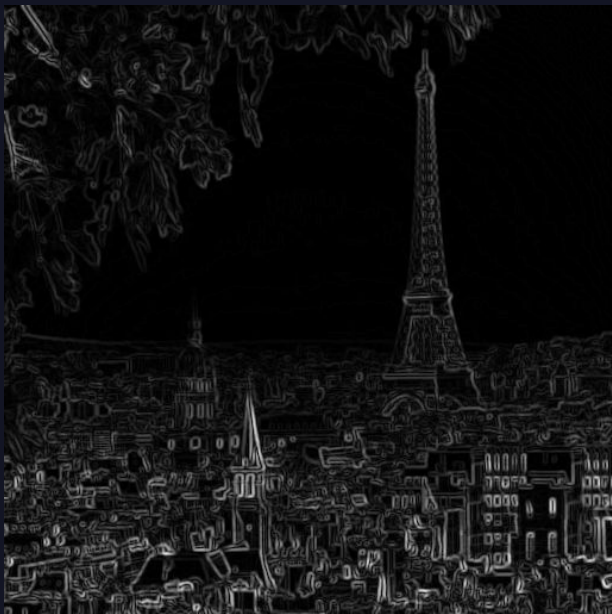
Espace couleur : Lab

Type de compression : palette



## 2 – SLIC et compression

### 2. Perturbation des centres des clusters grâce à un gradient



## 2 – SLIC et compression

### 3. Mise à jour des centres des clusters et attribution des couleurs



Itération n°1

PSNR = 26.0528 dB



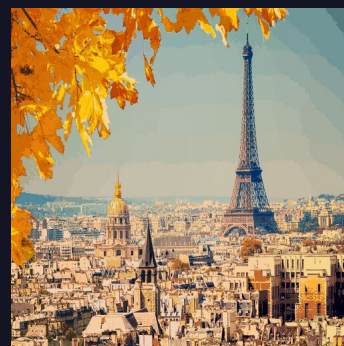
Itération n°2

PSNR = 26.2205 dB



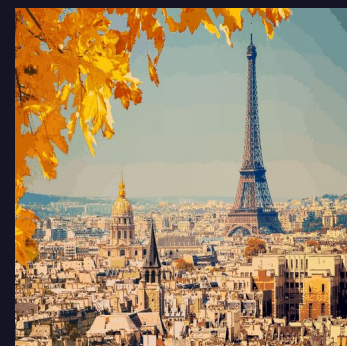
Itération n°3

PSNR = 26.4358 dB



Itération n°4

PSNR = 26.5779 dB



Itération n°5

PSNR = 26.6318 dB

Taux de compression = 2.61   Taux de compression = 2.63   Taux de compression = 2.65   Taux de compression = 2.68   Taux de compression = 2.68  
Entropie = 5.95 bits / pixels

# 3 – Résultats

Quelques résultats intéressants ( $\text{PSNR} \geq 30 \text{ dB}$ ) :

/	K	m	voisinage	Nbr iter	Espace couleur	PSNR (en dB)	Taux de compression	Entropie (bits/pixels)
image 1	150	10	3	5	Lab	30.095	2.724	6.89
image 2	280	1	3	5	Lab	31.648	1.781	6.53
image 3	1000	2	3	5	Lab	33.447	1.746	7.08
image 4	250	2	6	1	Lab	30.626	1.942	6.33



image 1



image 2



image 3



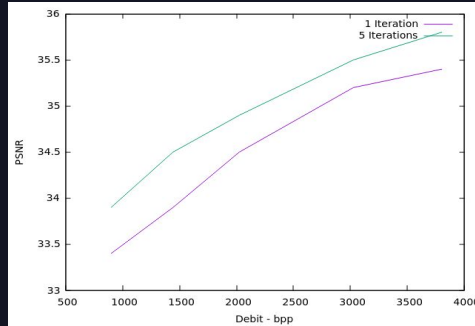
image 4

# 3 - Résultats

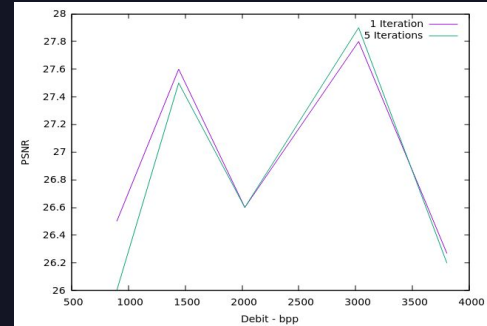
Quelques courbes :

Compacité (m) : 5

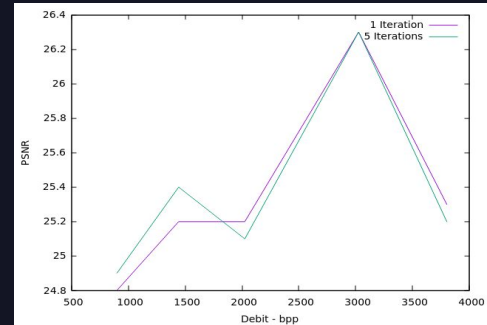
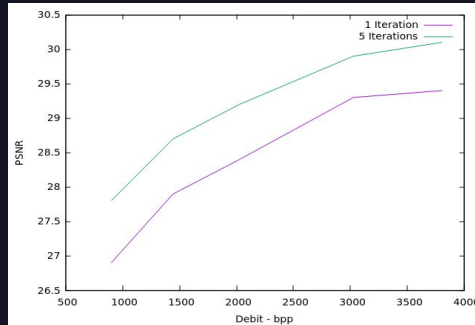
Sans voisinage du gradient



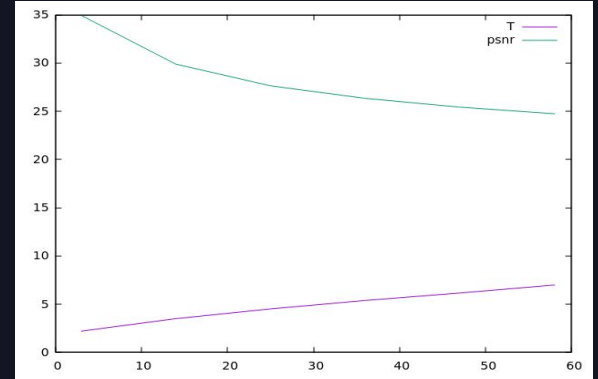
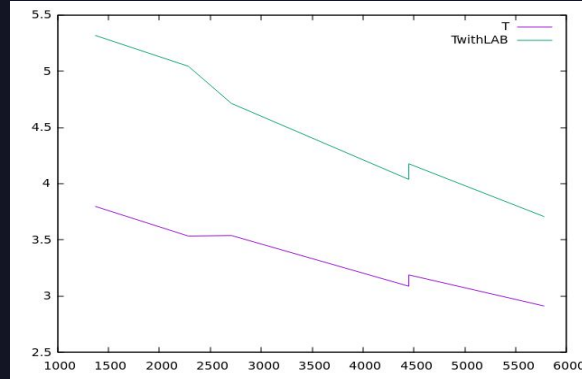
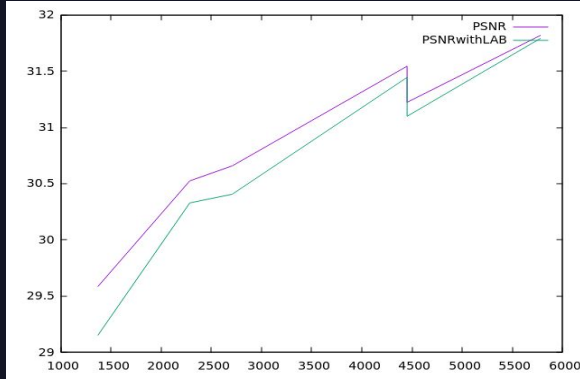
Avec voisinage du gradient



Compacité (m) : 30



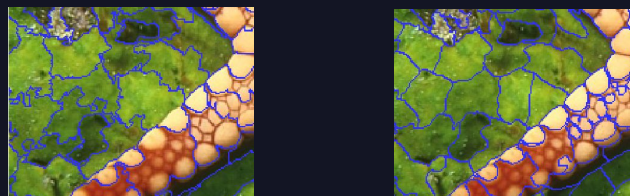
# 3 - Résultats





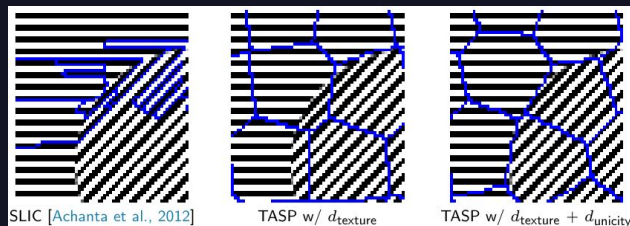
# 4 – D'autres algorithmes et nos tentatives

## TASP et SLIC&Merge :



SLIC

TASP



TASP est basé sur SLIC et s'appuie en plus sur un algorithme PatchMatch.

TASP = SLIC + Distance texture + Distance d'unicité

## Conditions du merge :

- . Superpixels adjacents
- . Même couleur

SLIC



K = 250 / m = 2 / PSNR = 31.049 dB

SLIC&Merge



K = 250 / m = 2 / PSNR = 30.6267

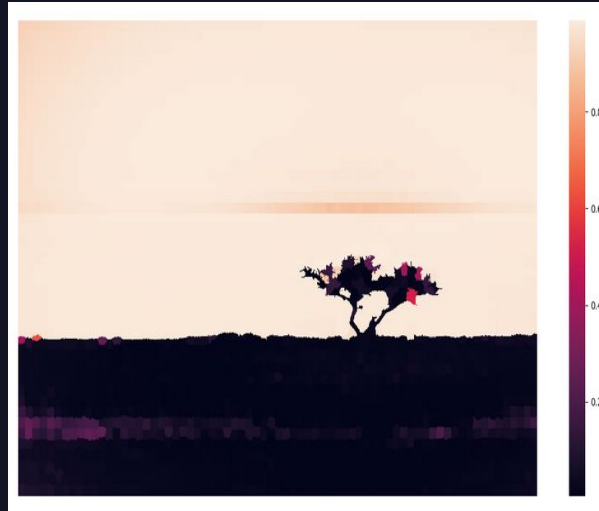
# 5 – D'autres possibilités d'utilisations

## Utilisation du deep et/ou du machine learning :

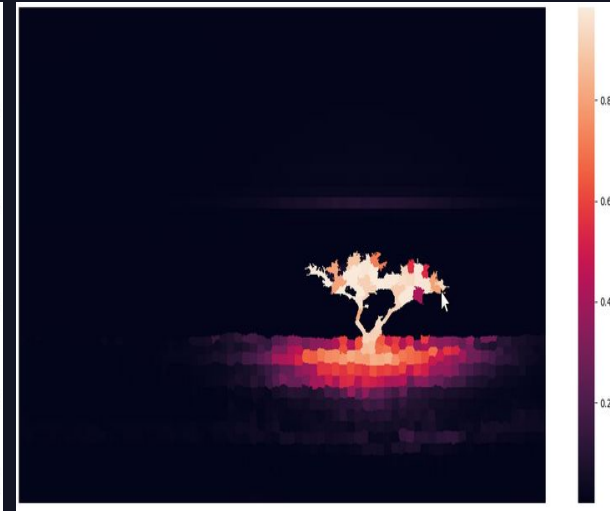
Exemple : Utilisation d'un séparateur à vaste marge



Image originale



Probabilité d'appartenir au ciel



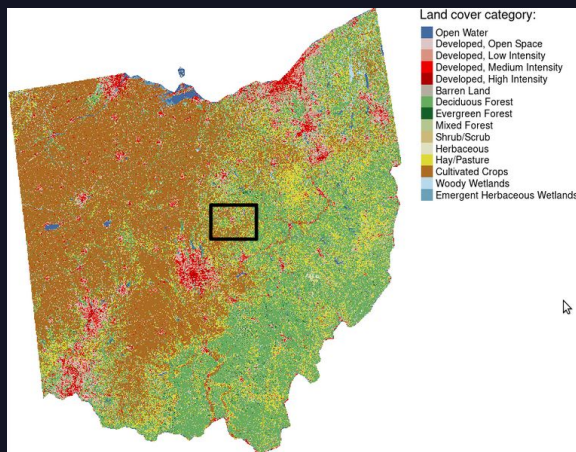
Probabilité d'appartenir à l'arbre



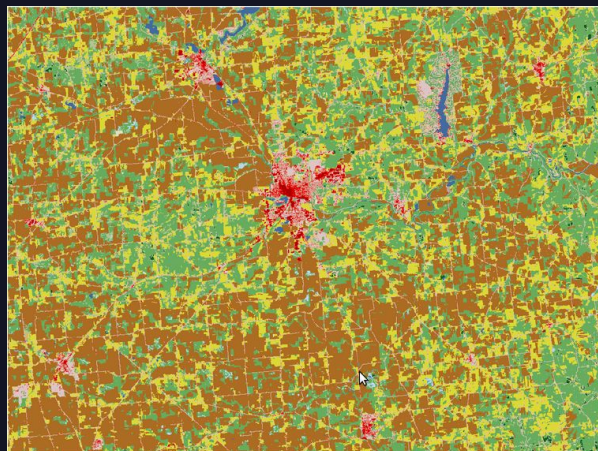
# 5 – D'autres possibilités d'utilisations

Utilisation de la segmentation en superpixels dans différents domaines :

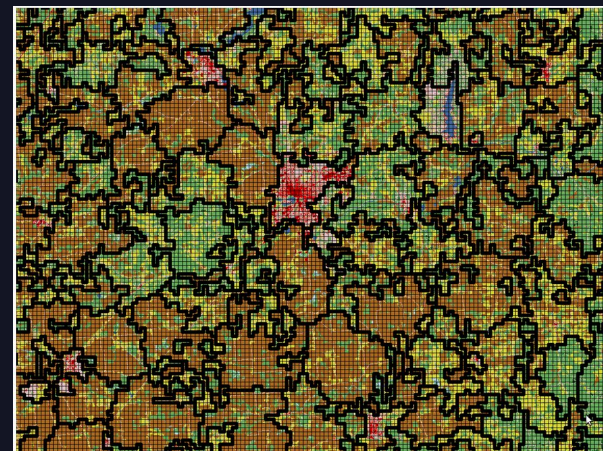
Exemple : Utilisation sur une “heatmap” représentant une partie de l'Ohio



Classification initiale

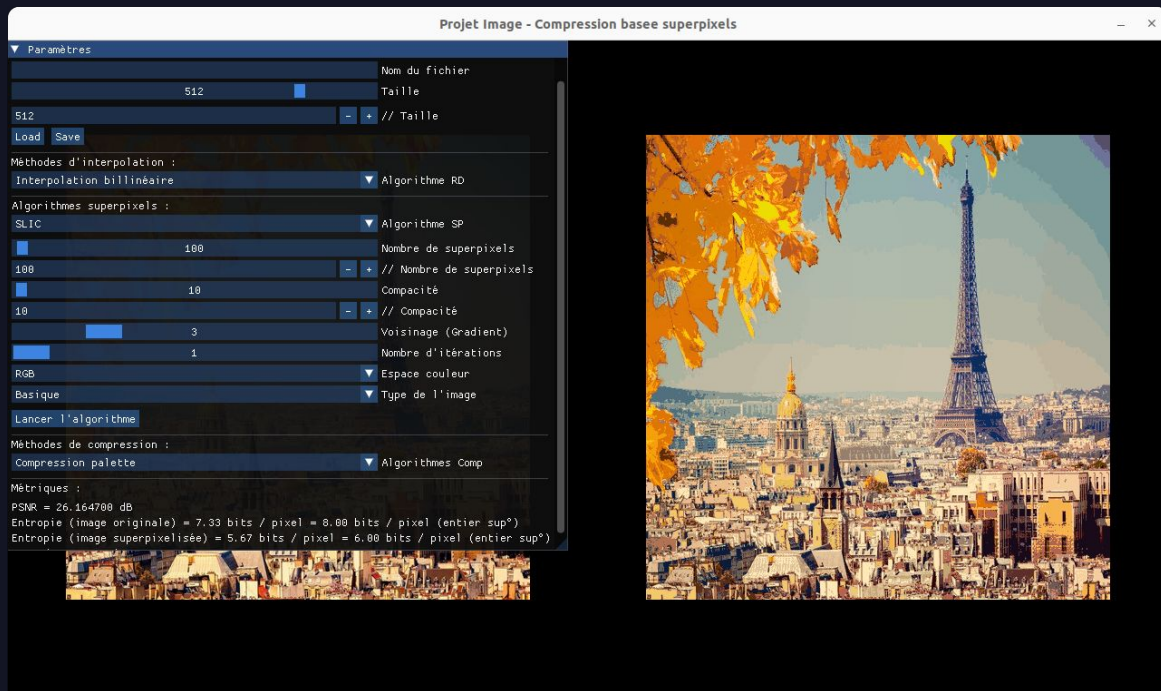


Classification du carré noir



Segmentation finale en superpixels

# 6 – L'interface graphique et démonstration



. Développé en C++

. Utilisation de la bibliothèque SDL2  
(pour l'initialisation de la fenêtre)  
ainsi que de ImGui (UI)

# Fin

**Merci d'avoir suivi notre présentation !**

**Avez-vous des questions ?**

# Références

<https://jakubnowosad.com/giscience-2021/#8>

[https://fr.wikipedia.org/wiki/L\\*a\\*b\\*\\_CIE\\_1976](https://fr.wikipedia.org/wiki/L*a*b*_CIE_1976)

[https://remi-giraud.enseirb-matmeca.fr/data/download/pres/Slides\\_TASP\\_ICIP\\_2019.pdf](https://remi-giraud.enseirb-matmeca.fr/data/download/pres/Slides_TASP_ICIP_2019.pdf)