

Projet GPGPU

Fall 2023 - GISTRE/SCIA S9

Objectif

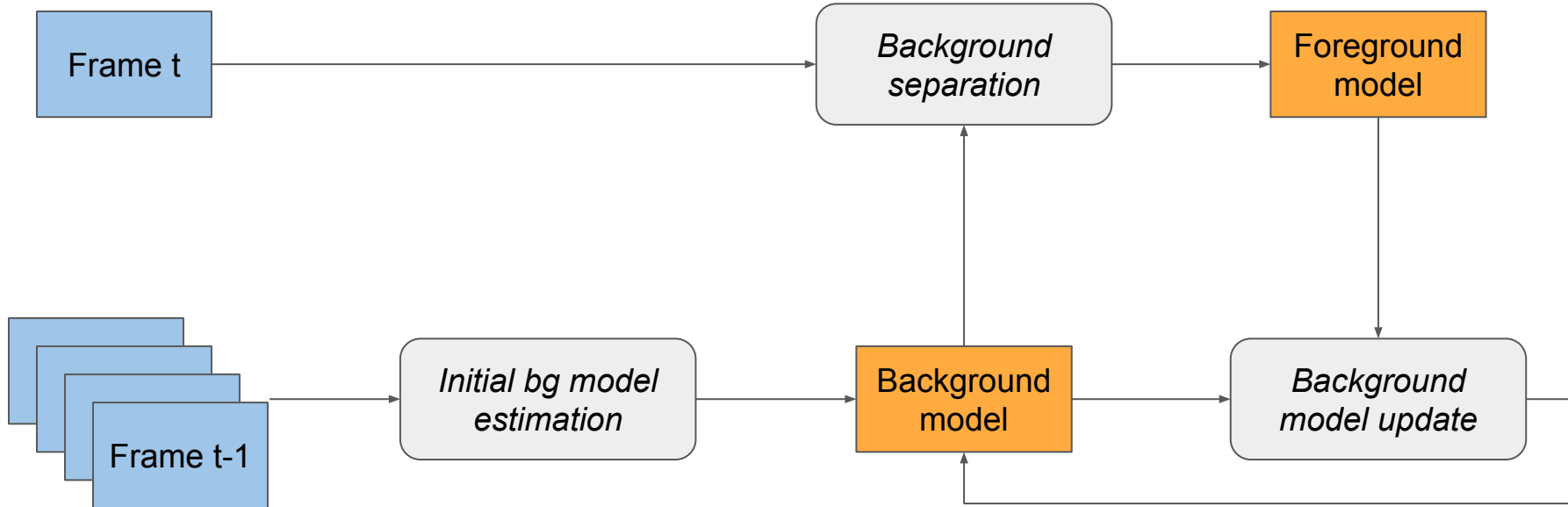
Plugin GStreamer de séparation fond / objets mobiles dans des vidéos

Étape préliminaire dans de nombreuses chaînes de traitement

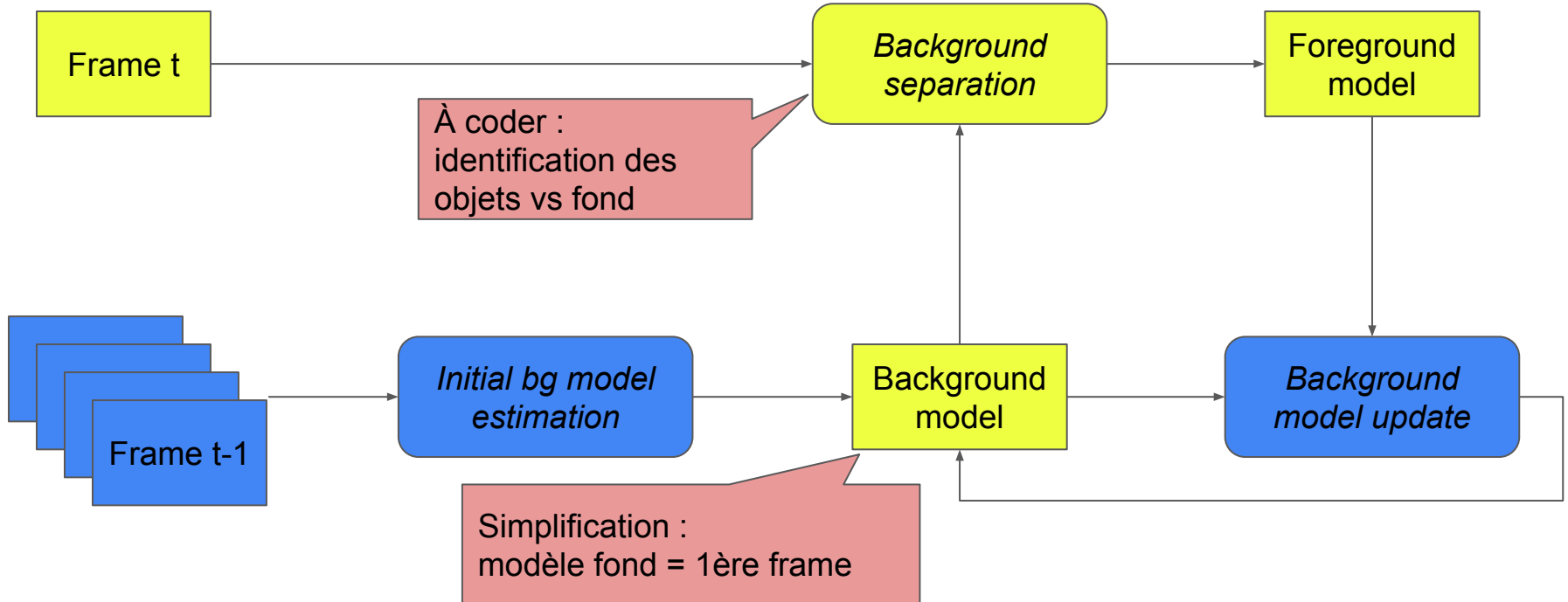
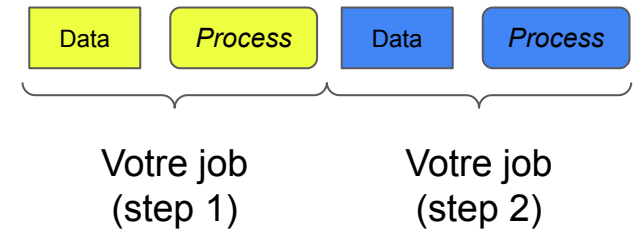
Beaucoup d'opération locales \Rightarrow Bon candidat pour une optimisation GPU



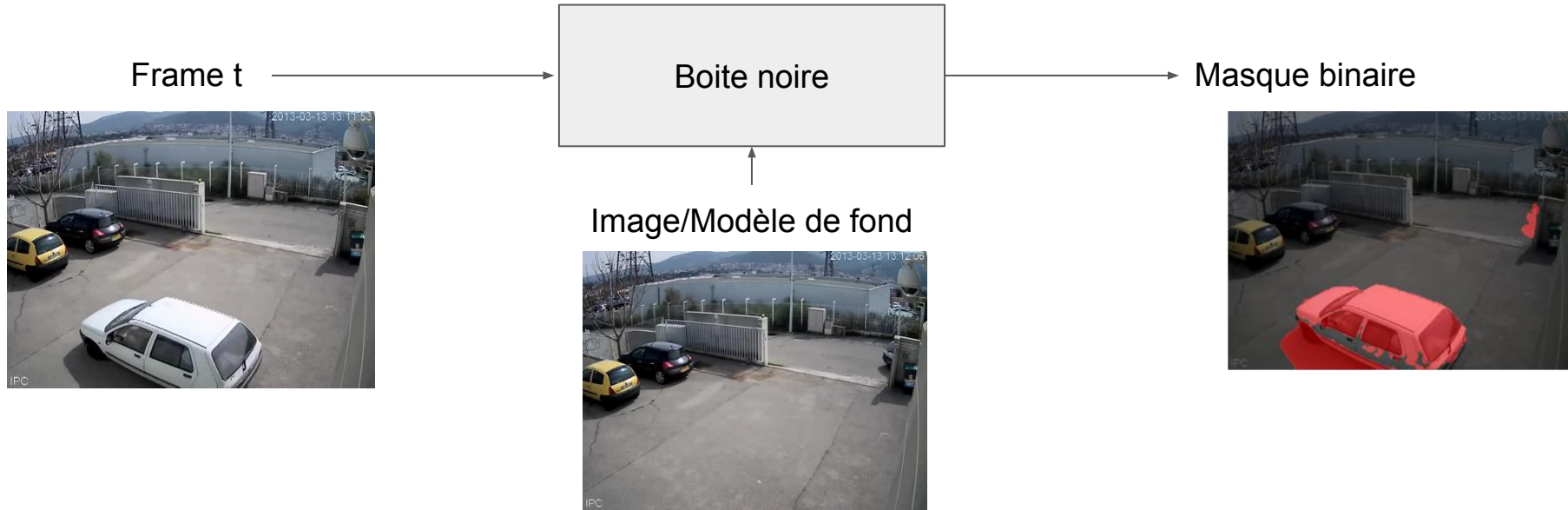
Approche générale



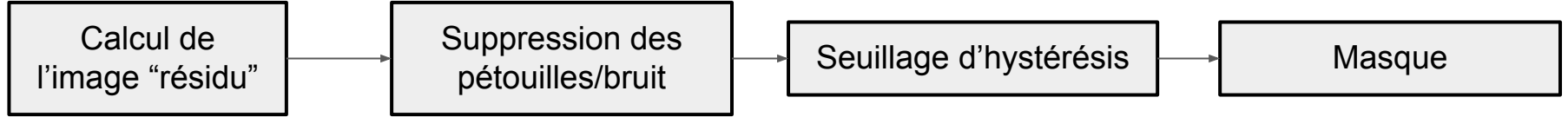
Approche pour le projet



Calcul du masque de changement



Détail des étapes



$$\Delta E_{ab}^* = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2}.$$

Distance dans l'espace
La*b*

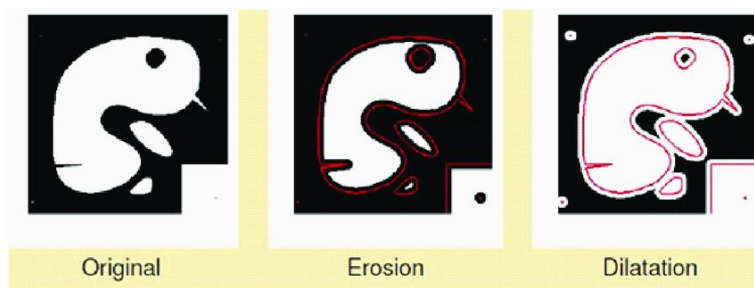
Ouverture morphologique
par un *disque* de rayon 3.

Seuillage d'hystérésis.
Seuil bas: 4
Seuil haut: 30

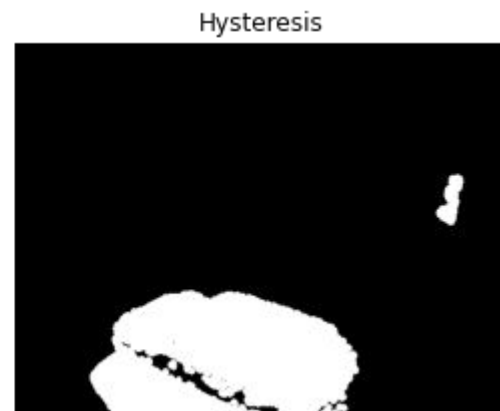
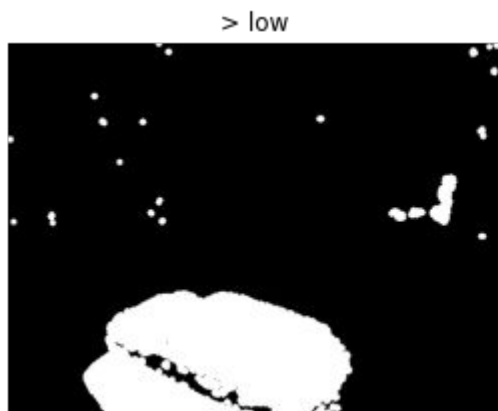
input + 0.5 * red * masque

Détail des étapes

Ouverture morphologique (erosion + dilatation)



Filtre de hysteresis



Implémentation de la reconstruction d'hystérésis

Idée:

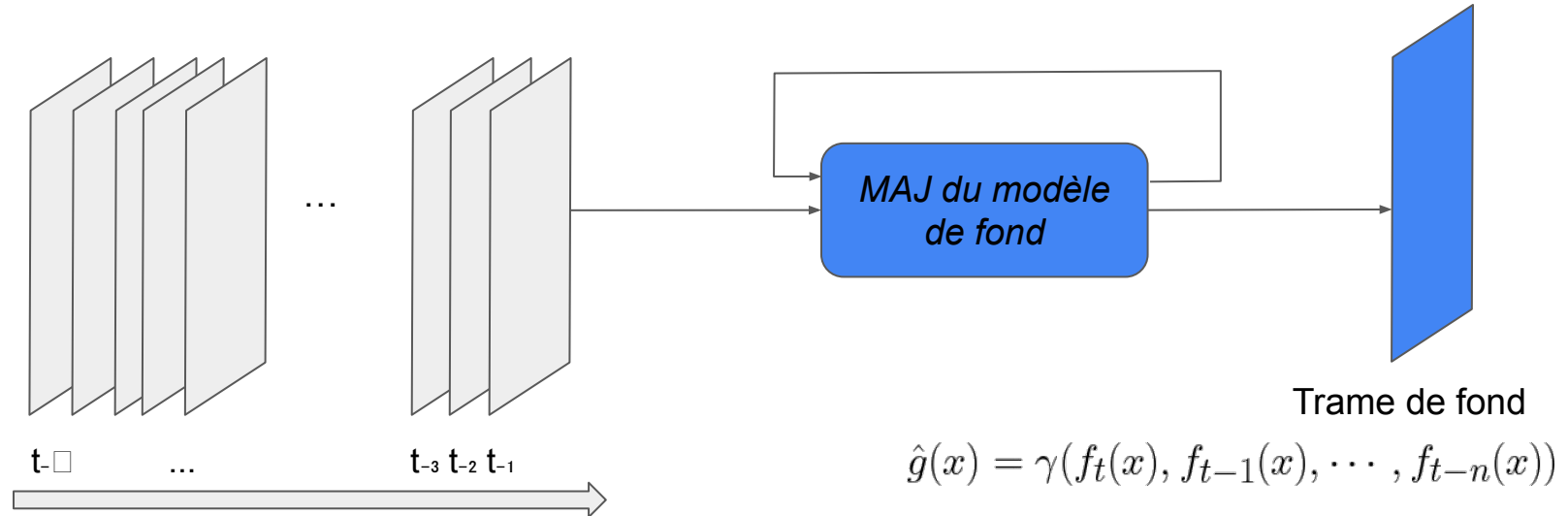
On propage les pixels des
markers dans le masque jusqu'à
stabilité.

```
__device__ bool has_changed;

__global__ void reconstruction(input, marker, out) {
    int p = .... + threadIdx.x;
    if (out[p] || !input[p])
        return;
    if (marker[p]) {
        out[p] = true;
        has_changed = true;
        return;
    }
    for (int q : neighbors(p))
        if (out[q]) {
            out[p] = true;
            has_changed = true;
        }
}

int main() {
    out = black_image
    Has_changed = false;
    while (has_changed)
        reconstruction<<<>>>(input, marker, out);
}
```


Estimation de la l'image de fond



Avec γ qui est:

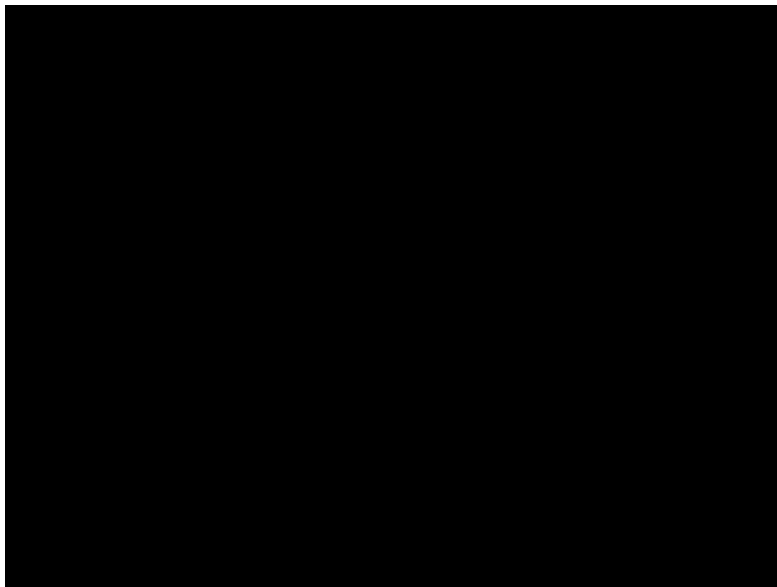
1. la moyenne (facile)
2. Le median (plus robuste mais plus d'ûr)

Code gstreamer

- Code de base fourni sur Moodle
- Vous devez implémenter un filtre GStreamer CUDA et CPP ayant pour paramètres:

```
bg=uri           : uri vers une image de fond (default="" => estimé)
opening_size=(int) : taille de l'ouverture
th_low=(int)      : valeur basse du filtre (default=3)
th_high=(int)     : valeur haute du filtre (default=30)
```

```
bg_sampling_rate=(int) : intervalle d'échantillonnage des frames pour l'estimation de fond
                        (default=500ms)
bg_number_frame=(int)  : nombre de frames utilisées pour l'estimation de fond (default=10)
```



```
gst-launch-1.0 uridecodebin uri=file:///tmp/sintel_trailer-480p.webm ! videoconvert ! "video/x-raw,  
format=(string)RGB" ! cudafilter ! videoconvert ! video/x-raw, format=I420 ! x264enc ! mp4mux ! filesink  
location=video.mp4
```

Demo

Critères d'évaluation

1. **Code correct** \Rightarrow résultats ACCEPTABLES au niveau qualitatif
Avec cette méthode, les résultats ne seront pas optimaux
2. **Vitesse** \Rightarrow plus le framerate est rapide, mieux c'est.

Conseils

1. **Avoir une version C++ fonctionnelle** \Rightarrow baseline
2. **Git tag des versions du programme** \Rightarrow permet de mesurer plusieurs versions/optimisations du programme
3. **Faire les optimisations une par une** \Rightarrow permet d'évaluer les optimisations réellement significatives de façon individuelle

Livrables

1. Implémentation

- Source code for C++ CPU reference
- Source code for CUDA implementation(s)
- Source code for benchmark tools
- Build scripts (GNU Make, CMake...)

Nous devons être capables de reproduire vos résultats

2. Rapport succinct

- Description du sujet
- Répartition des tâches par membre du groupe
- Benchmarks et graphiques des performances des versions (CPU + GPU + GPU Optimisé ##)
- Analyse des performances et des bottlenecks (graphiques nsight / nvprof)

3. Slides de la soutenance

4. Répartition des groupes (sur Moodle) => 31 octobre (pour les affectations de soutenance)

Soutenances

1^{er}
décembre

- 15' présentation
- 5' démo
 - *Données:* <https://cloud.lrde.epita.fr/s/xZXGCa8dMX3GsTW>
- 5' discussion

Soutenances sur Teams.

Projet par groupe de 4.

Tous les membres du groupe doivent être présents à la soutenance.

Vous devez rendre tous les fichiers la veille au soir et être inscrit·e à votre groupe.