Changement d'apparence du visage accéléré par CUDA et déporté sur e-GPU

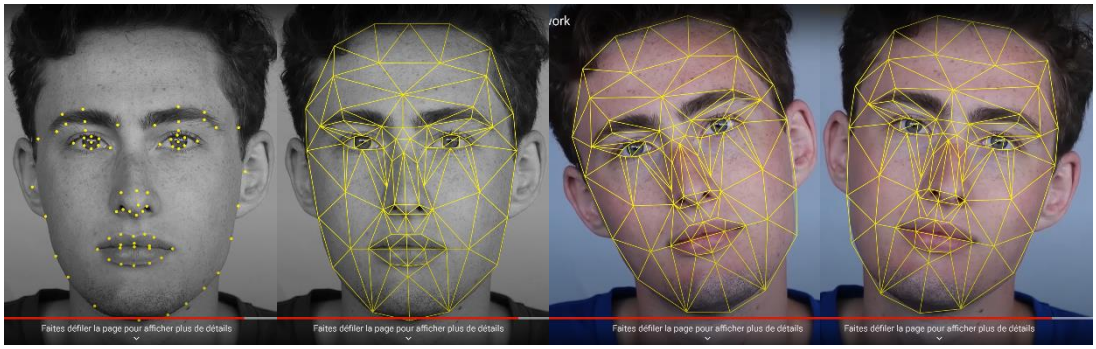
Mots-clés : GPU/e-GPU, CPU, CUDA, calcul parallèle, détection de visage, transformation d'apparence

Contexte

De nombreux algorithmes de traitement du signal peuvent être fortement accélérés en exécutant les calculs de manière parallèle sur les processeurs graphiques (GPU) présents sur les ordinateurs, les tablettes ou encore les smartphones. Le plus souvent utilisée dans le contexte de jeux ou de CAO/DAO, la puissance de calcul offerte par les GPU est aujourd'hui supérieure à celle des processeurs classiques en raison d'un nombre de transistors bien plus important et d'architectures spécifiques fortement parallélisées. Le leader du secteur NVIDIA a développé CUDA, une architecture à la fois matérielle et logicielle, qui permet d'implémenter de manière très simple des algorithmes parallélisables et de mêler, si nécessaire, calcul, rendu graphique et interaction.

Descriptif

D'une manière générale, la détection automatique de visage trouve de nombreuses applications. A travers la simple détermination dans une scène d'une ou plusieurs régions rectangulaires approximatives représentant chacune potentiellement un visage, elle concerne la détection d'intrusion ou de présence, le dénombrement du nombre d'individus, l'analyse comportementale d'une clientèle ou d'une population, etc. A partir d'une base de données regroupant les empreintes visuelles d'individus, un certain nombre de méthodes permettent ensuite la reconnaissance pour l'identification ou l'accès à une ressource. La détermination de la posture ou de points remarquables de visages (voir figure ci-dessous) rendent possibles des changements d'apparence plus ou moins complexes : transformations géométriques ou ajout d'accessoires ludiques dans des applications de réseaux sociaux, remplacement d'acteurs réels ou virtuels dans des scènes vidéos, simulations de changement d'âge, anonymisation par incorporation de visage « flouté » ou « neutre », etc.



L'objectif du projet est d'utiliser l'architecture CUDA pour accélérer à la volée, en temps-réel, des algorithmes de changement d'apparence du visage à des fins d'anonymisation (« floutage » passe-bas ou aléatoire) ou ludiques (filtres colorimétriques, géométriques, ajout de nouveaux éléments tels qu'un masque). La démarche est basée sur une détection automatique préalable du visage permettant d'identifier sa position, sa posture ou sa géométrie à travers des points remarquables. A cette fin, la bibliothèque de traitement d'images OpenCV qui inclut un algorithme détectant l'intégralité des visages présents dans une image sous la forme d'une liste de régions pourra être utilisée. OpenCV permet également la réalisation simple d'interfaces minimalistes fenêtrées ainsi que la gestion de flux vidéos temps-réel issus de caméras. Les calculs de changement d'apparence, réalisés sous CUDA,

devront être intégrés à l'interface et mis en œuvre en Langage C. Une performance supplémentaire sera obtenue en mettant à profit des calculs accélérés sur un boitier e-GPU Razer Core X Chroma embarquant une carte NVIDIA RTX 3080. Selon l'état d'avancement, afin d'obtenir une vitesse d'exécution optimale, l'implémentation sous CUDA pourra concerner l'intégralité des traitements incluant également la détection automatique de visage.


Le projet se décomposera en plusieurs volets séquentiels ou parallèles :

1. Etude bibliographique portant sur le calcul scientifique par GPU à l'aide de l'architecture CUDA, les applicatifs ou les bibliothèques déjà existantes, les API disponibles.
2. Etude bibliographique portant sur les algorithmes de détection de visages (régions ou points caractéristiques, sur les implémentations existantes et sur les méthodes de changement d'apparence.
3. Prise en main de la bibliothèque OpenCV en Langage C et de la gestion de flux vidéo temps-réel.
4. Implémentations GPU (CUDA sur stations de travail, ordinateurs portables ou sur e-GPU) mettant en oeuvre quelques algorithmes de changement d'apparence et mesure des performances obtenues.

Liens connexes

« How Snapchat's filters work », https://www.youtube.com/watch?v=Pc2aJxnmzh0&feature=youtu.be
« OpenCV – Cascade Classifier », https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
« OpenCV – Face Filters », https://github.com/kunalgupta777/OpenCV-Face-Filters
« Programming Snapchat-Like Filters », https://dev.to/unqlite_db/progamming-snapchat-like-filters-cod
« Libface detection », https://www.developpez.com/actu/251293/Libfacedetection-une-bibliotheque-Cplusplus-open-source-de-detection-de-visages-permet-en-une-seconde-de-gerer-1500-images-de-128-pixels-par-96-pixels/
« Détection faciale et reconnaissance faciale avec OpenCV4 en C++ », https://m-applications.devoteam.com/2019/02/11/detection-faciale-et-reconnaissance-faciale-avec-opencv4-en-c/
« La détection des visages – OpenCV », https://open.mit.edu/c/workofthefuture/1gv/la-d%C3%A9tection-des-visages-opencv-by-meissa-rassoul

Contact

Encadrant : Marc Donias
Bureau : B2.35 (IMS)
Téléphone : 05 40 00 84 18
Courriel : marc.donias@enseirb-matmeca.fr