

Projet 4ETI - partie IMI

CPE

4ETI IMI

Ce projet vous demande de mettre en oeuvre des compétences en traitement et/ou synthèse d'images dans le cadre d'un sujet libre.

Lors de ce projet, vous aller vous familiariser avec la bibliothèque [OpenCV](#) (Open Source Computer Vision) pour l'analyse d'images. Cette bibliothèque, répandue dans le domaine de l'analyse d'images est particulièrement spécialisée pour des applications en vision par ordinateur¹ (Computer Vision en anglais) correspondant à la détection d'objets/de personnes, au suivi de mouvement, au tracking, etc.

OpenCV propose une API en C++ par défaut ainsi qu'en Python pour développer des prototypes. De part sa structure en C++, OpenCV propose une interface de plus bas niveau que Matlab, mais permet à l'opposé la mise en oeuvre d'algorithmes rapides et efficaces. En particulier, il peut être utilisé pour de la détection et du traitement d'images en temps réel sur un flux vidéo. Notez que la bibliothèque OpenCV est sous licence BSD, elle est à ce titre utilisée à la fois dans des projets open sources tout comme dans des projets commerciaux en entreprise.

Contents

1	Affichage de la webcam	2
2	Analyse d'image dans une scène OpenGL	2
3	Affichage d'un objet 3D dans une image	2
4	Sujet du projet	3
4.1	Choix du sujet	3
4.1.1	Exemples de sujets possibles	3
4.2	Remarques générales sur ce qui est attendu	3
4.3	Remarque sur l'utilisation et la réutilisation de code	3
5	Rendu	4
5.1	Format et contenu de l'archive	4
5.2	Contenu du rapport	4
5.3	Plan suggéré du rapport	5
5.3.1	Étude de l'existant	5
5.3.2	Validation de l'approche	5
5.3.3	Analyse des résultats	6
5.3.4	Explication des limitations	6
5.4	Inclusions d'images	6
5.4.1	Format d'images	6
5.4.2	Schémas explicatifs	6

¹Notez également la bibliothèque [ITK](#) dans le cas de l'analyse d'images médicales

1 Affichage de la webcam

Question 1 *Compilez et exécutez le premier code.*

Ce code récupère dans une boucle les images issues de la webcam et les affiche dans une fenêtre. L'image est récupérée dans une classe `Mat` interne à OpenCV.

Notez qu'à partir de ce code minimaliste, il est possible d'appliquer des analyses et transformations d'images de manière efficace sur un flux vidéo en s'aidant des fonctions d'OpenCV et/ou en codant de nouvelles.

2 Analyse d'image dans une scène OpenGL

Question 2 *Compilez et exécutez le second code.*

Ce code permet d'interfacer OpenCV et un affichage 3D avec OpenGL. Cette fois, l'image de la webcam est récupérée par le biais d'OpenCV et est transférée sous forme de texture à la carte graphique par le biais d'OpenGL qui réalise ensuite son affichage sur des triangles dans une scène 3D.

Question 3 *Placez devant la caméra un objet suffisamment large de couleur rouge (pochette, t-shirt, etc). Que ce passe-t-il ? Observez dans le code du projet comment est géré cette interaction (fichier `scene.cpp`).*

Nous sommes désormais en face d'une scène 3D interagissant avec le monde extérieur. Ce type de comportement est caractéristique de ce que l'on désigne par **Réalité Virtuelle**. L'utilisation d'OpenCV pour la partie d'analyse d'images associée à OpenGL pour la partie visualisation 3D (et potentiellement traitement d'images dans les shaders) permet de mettre en place des systèmes de réalités virtuelles en temps réel.

3 Affichage d'un objet 3D dans une image

Question 4 *Compilez et exécutez le troisième code qui affiche par défaut uniquement l'image de la webcam sur l'écran.*

Question 5 *Placez devant la caméra un objet suffisamment large de couleur rouge (pochette, t-shirt, etc). Que ce passe-t-il ? Observez dans le code du projet comment est géré cette interaction (fichier `scene.cpp`).*

Notez que la scène qui semble 2D à l'origine est en fait constitué d'un carré s'affichant en plein écran dans une scène OpenGL. Il est alors possible d'ajouter n'importe quel élément 2D ou 3D en manipulant la scène OpenGL.

Cette fois, nous intégrons des informations supplémentaires (ici 3D) à partir d'une image réelle. Ce type d'ajout dépendant d'une image visionnée est désignée sous le terme de **Réalité Augmentée**.

Encore une fois, l'utilisation d'OpenCV pour la partie image, et OpenGL pour la partie réalité augmentée permet d'avoir une interaction temps réelle avec l'intégration d'éléments 3D en interaction avec une image réelle.

4 Sujet du projet

4.1 Choix du sujet

Pour ce projet, vous êtes libre de choisir par binôme ou trinôme un sujet que vous allez développer.

Ce sujet pourra consister soit

- En une analyse et traitement d'images sur un flux vidéo (voir code 1).
- En un projet de réalité virtuelle (voir code 2).
- En un projet de réalité augmentée (voir code 3).

4.1.1 Exemples de sujets possibles

(Vous êtes libres d'en trouver d'autres)

- Détection faciale.
- Tracking de personne.
- Contrôle d'un objet 3D à partir de caractéristiques d'une image (couleur, forme).
- Contrôle d'un objet 3D à partir de la posture du corps ou du mouvement.
- Interaction avec une forme 2D ou 3D à partir de la webcam (jeu de pong, casse brique *manuel*, etc)
- Affichage d'informations sur des objets vus par la webcam.
- etc.

4.2 Remarques générales sur ce qui est attendu

- Il n'est **pas** indispensable d'utiliser OpenGL si vous ne souhaitez pas interfacier votre scène avec des objets 3D. Il n'est pas indispensable d'utiliser OpenCV non plus, cependant il n'est pas autorisé de faire entièrement un projet sous Matlab (sauf cas exceptionnel justifié et demandé clairement).
- Il est désormais de **votre ressort** de **comprendre** l'utilisation et les fonctionnalités d'OpenCV et de les **mettre en oeuvre** à partir de vos connaissances et des tests que vous pourrez réaliser.
- Le but du projet est **libre**, mais doit faire ressortir une **démarche d'analyse scientifique**.

4.3 Remarque sur l'utilisation et la réutilisation de code

Il existe de nombreuses fonctions toutes faites pour OpenCV et de très nombreux codes sur internet réalisant des traitements d'images plus ou moins avancé.

Le but du projet n'est pas d'utiliser simplement une fonction toute faite d'OpenCV ou d'intégrer un code qui réalise une détection complexe. Si vous utilisez une fonction toute faite, ou si vous réutilisez un code existant trouvé sur internet, alors

- Il faut le **dire clairement** (dans le code et dans le rapport), attention à ne pas tomber sur un cas de plagiat qui aurait des conséquences potentiellement graves (rappel de 3ETI).

- Il faut que vous **comprenez** ce que le code réalise et que vous **expliquiez** la méthode que vous utilisez dans votre rapport (ex. Détection faciale par *eigenface*, etc.) afin de montrer que vous avez compris les outils que vous utilisez.

La notation tiendra compte à la fois du résultat de votre projet, mais également de la manière dont vous restituerez ce que vous aurez compris des outils que vous utilisez. Un projet aux résultats impressionnants utilisant des outils complexes, mais insuffisamment maîtrisés ne sera pas forcément mieux noté qu'un projet plus simple mais mieux maîtrisé.

5 Rendu

5.1 Format et contenu de l'archive

Vous rendrez votre projet en tant qu'archive numérique avant la fermeture du dépôt du e-campus sous la forme suivante:

- Votre archive sera sous la forme `tar.gz`.
- Votre archive sera nommée sous la forme `nom1_nom2.tar.gz`. (voir `nom3` dans le cas d'un trinome).
- Votre archive contiendra
 - Obligatoire: Une image représentative de votre projet (au moins une).
 - Obligatoire: Votre code (uniquement les fichiers sources et les données nécessaires à son exécution, pas les fichiers temporaires de compilations ou de votre IDE).
 - Obligatoire: Un rapport au format `.pdf` décrivant votre projet.
 - Optionnel: Une vidéo représentative de votre projet.

Vous rendrez également la webcam qui vous a potentiellement été confiée. Notez que votre projet sera considéré comme **incomplet** tant que le matériel mis à votre disposition n'aura pas été rendu.

5.2 Contenu du rapport

L'écriture de votre rapport sera l'occasion de vous entraîner à rédiger un rapport **scientifique et technique** en vue de la préparation de ceux attendu en fin de stage (type PFE) ou pour ceux attendus dans votre carrière afin de présenter une solution technique de manière scientifique (rapport pour un supérieur, explication d'un choix technique, etc).

Notez que les différentes remarques et conseils ci-après sont valables pour votre rendu de projet, mais également pour tout autre rendu scientifique et technique à la fois en entreprise et en centre de recherche. La plupart des conseils sont valables également pour les présentations, pas uniquement pour les rapports.

Tout comme en PFE, un **rapport de qualité insuffisante** (tout en étant associé à un projet valide) devra être **refait** et comptera en seconde session.

Votre rapport devrait faire entre 8 et 15 pages (notez qu'un rapport de PFE doit faire 30 pages sans les annexes).

5.3 Plan suggéré du rapport

1. Présentation du contexte, de l'objectif, de la problématique.
2. Étude de l'existant (différentes approches possibles pour résoudre ce problème). \triangle
 - Explication des avantages et inconvénients de chaque approche. \triangle
 - Comparaisons entre les approches. \triangle
3. Explication de l'approche choisie et de sa mise en place (cette partie de méthode se décline en fait en autant de parties que nécessaire dans votre description).
4. Résultats obtenus et analyse de ceux-ci.
 - Validation de l'approche (= résultats obtenus sur des cas caractéristiques reproductibles et quantifiés). \triangle
 - Observation et **Analyse** \triangle des résultats (analyse sur des cas d'applications finales).
5. Explication des limitations. \triangle
6. Conclusion.

Notez les \triangle correspondants à des points généralement oubliés ou classiquement insuffisamment traités dans vos rapports.

Notez également qu'il est de votre responsabilité d'adapter ce plan à votre projet (choix des titres, nombres de parties et sous-parties, etc).

5.3.1 Étude de l'existant

Étude de l'existant. Il est toujours très important de lister et d'analyser les différentes approches possibles permettant de répondre à un problème. Cela permet de justifier vos choix et de comprendre son avantage et ses limitations. Ne commencez pas directement à décrire votre approche sans avoir expliqué pourquoi vous choisissez celle-ci et pas une autre.

Notez qu'en entreprise, quand bien même le choix serait imposé, il est de votre responsabilité d'identifier d'autres approches possibles et de les analyser.

Notez également que l'approche que vous choisissez n'est pas forcément sans défauts ni limitations, mais expliquez clairement (ex. tableau comparatif) pourquoi il s'agit de l'approche choisie dans votre cas particulier en prenant clairement conscience des limitations.

5.3.2 Validation de l'approche

Montrer un résultat dans un cas final n'est pas suffisant. Il faut montrer que votre approche répond au problème sur des cas minimalistes bien identifiée (ex. détection dans le cas d'un cercle parfait sans bruit, analyse de la même image que celle utilisée dans la base de données, etc).

C'est à vous de définir des critères de bases qui permettent de valider les bases de votre méthode. Il est possible de voir cette validation comme le fait de passer des tests de bases (similairement à des tests unitaires sur un code). Le choix de ces tests fait partie de votre travail d'ingénieur. Ces tests doivent être les plus quantifiés possible (nombres de pixels, temps de calcul, quantités mémoires, etc). N'indiquez pas juste *nous avons testés sur un rond rouge*. Notez qu'il est possible que votre méthode finale échoue sur un cas de base prévue en début de projet. Ne le cachez pas, montrez bien que vous avez pris cela en compte pour de futurs travaux.

5.3.3 Analyse des résultats

Une erreur classique consiste à uniquement montrer des résultats sans les analyser. Que tirez vous comme conclusion de ces résultats? En particulier, évitez absolument d'indiquer ce type d'analyse: *Nous pouvons constater que la méthode marche bien*. En quoi la méthode *marche* bien ? Avez-vous testé sur un jeu de données différent, sous différents éclairages, avez-vous tracés des courbes quantifiant la qualité de vos résultats. Quelle marge d'erreurs avez-vous? Est-ce dépendant de l'orientation, position, etc. Notez qu'une analyse n'est pas toujours positive, elle peut mettre également en avant les limitations de la méthode.

5.3.4 Explication des limitations

Un projet scientifique de qualité montre à la fois les points positifs de la méthode réalisée, mais également ces limitations. Soyez honnête sur la méthode réalisée, et expliquez-les clairement les cas que l'approche ne peut pas traiter. Expliquez pourquoi (et ne montrer pas uniquement un exemple ne fonctionnant pas sans donner de raisons). Expliquez également les pistes possibles permettant de corriger ces limitations en expliquant les hypothèses supplémentaires à prendre en compte ou en quantifiant le coût (en temps de calcul, de développement, voir en argent) que cela engendrerait.

5.4 Inclusions d'images

Notez que pour un rapport traitant d'une problématique en image, il est **indispensable** que votre rapport contienne des images. Notez que vos images doivent être référencées dans le texte par un numéro et qu'une légende doit y être apportée.

5.4.1 Format d'images

N'incluez pas d'images non compressées dans votre document (format bmp, ppm, etc à proscrire). Préférez les formats .jpg pour des images réelles, et .png pour des images de synthèses et des schémas. Adaptez également la taille de vos images dans un format raisonnable. Il est inutile d'inclure une image de 2500 × 2500 pixels dans une petite case de votre rapport. Redimensionnez chaque image (600x600 généralement suffisant pour la plupart des images ne comportant pas de texte). Assurez vous que votre image soit lisible pour une lecture standard et n'allez pas au delà. Vérifiez régulièrement la taille de votre rapport. Celui-ci ne devrait pas dépasser 5 à 10Mo maximum. Inversement, n'incluez pas d'image de très mauvaise qualité ou trop petite comportant du texte qui deviendrait alors illisible. (Notez que la commande `convert` d'ImageMagick ou le logiciel Gimp peut permettre de réaliser des conversions et des redimensionnements d'images.)

5.4.2 Schémas explicatifs

Dans de nombreux cas, il est plus simple de décrire une approche ou un algorithme avec l'aide d'images et de schémas que par du texte uniquement. N'hésitez pas à faire des schémas descriptifs qui aident à la compréhension. Si un paragraphe de description dépasse 10 lignes, cela peut indiquer qu'un schéma serait plus adapté. Réaliser des schémas descriptifs de qualité est un travail parfois long (notez par exemple le logiciel Inkscape permettant la réalisation de schémas), si cela vous prend trop de temps, n'hésitez pas à réaliser un schéma à la main sur une feuille de papier et à inclure celui-ci en tant qu'image scannée dans votre rapport.