Domaine d'Approfondissement IMA
IMA 4102 | Analyse d'Image

LABS
Détection de Points d'Intérêt

Ce TP s'effectue en **binôme**.

Les **compte-rendus** sont à déposer sur **Moodle**
> moodle.imtbs-tsp.eu/course/view.php?id=636 > section **Livrables**
Les images résultats peuvent être incluses dans le document ou fournies séparément.

Date limite de remise
Groupe 1 > 26 mars 2024 à 23:00 | Groupe 2 > 27 mars 2024 à 23:00

Les **données** de ce TP sont disponibles sur **Partage**
> partage.imt.fr/index.php/s/6BbfW5pRsa4E4mG

- Les notebooks Python (`OpenCV_UneApplication.ipynb`) utilisés dans ce TP se trouvent dans le répertoire `Notebooks`
  avec `UneApplication = KeypointDetection | VideoKeypointDetection`
- Les images de test se trouvent dans le répertoire `Images`
- La documentation en ligne d'OpenCV est accessible à l'adresse docs.opencv.org/4.9.0
  Pour accéder à la documentation d'une routine, tapez son nom dans la fenêtre de recherche à droite sur la barre menu de la page courante.

A titre informatif, une implémentation C++ des codes Python associées est également disponible dans le répertoire `C++`

- Les sources OpenCV (`OpenCV-UneSolution.cpp`) se trouvent dans le répertoire `Sources`
  avec `UneSolution = KeypointDetection | VideoKeypointDetection`
- Les solutions Visual Studio (`UneSolution.sln`) résident dans les répertoires `UneSolution`
- Les exécutables associés (`UneSolution.exe`) sont générés dans les sous-répertoires `UneSolution/x64/Release` et ont été regroupés dans le répertoire `bin`

  Leur appel s'effectue en ligne de commande à partir d'un interpréteur Windows (`cmd.exe` | `powershell.exe` | `wt.exe`). Un appel sans arguments renvoie un message précisant la syntaxe.

Ces codes sont fournis dans le but :

1. d'illustrer la similarité entre les API Python et C++ d'OpenCV, qui rend aisée la transposition d'un code d'un langage à l'autre ;
2. de permettre aux étudiants susceptibles d'utiliser l'API C++ d'OpenCV dans le cadre de leur projet Cassiopée de disposer de codes modèles.

`OpenCV_KeypointDetection.ipynb` et `OpenCV_VideoKeypointDetection.ipynb` réalisent une détection de points d'intérêt, respectivement sur des images fixes et des séquences vidéos issues de la caméra courante, en utilisant les détecteurs de Lukas-Kanade-Tomasi (KLT) ou de Harris-Förstner :

- La routine `application()` de `OpenCV_KeypointDetection.ipynb` peut être appelée avec 1 ou 2 images d'entrée. Appliquée à 2 images de la même scène acquises dans des conditions et/ou sous des points de vue différents, elle permet d'apprécier la répétabilité des détecteurs.
- Ces derniers sont implantés via la routine OpenCV `goodFeaturesToTrack`.
- Leurs hyperparamètres sont ajustables interactivement à partir de l'interface graphique (GUI) :

GUI	Hyperparamètre	Valeurs
KLT | Harris	Détecteur	0 : KLT | 1 : Harris-Förstner
10*LScale	Echelle locale σ [noyau gaussien de variance σ^2]	10σ
IScale	Echelle d'intégration D [noyau unité de taille $(2D+1) \times (2D+1)$]	D
100*k	Paramètre α du détecteur de Harris	100α
Min dist	Distance euclidienne minimale d_{min} entre 2 points	d_{min}
Max #pts	Nombre maximal de détections N_{max}	N_{max}
QLevel	Seuil de détection défini comme la fraction $\gamma \in [0,1]$ de la réponse maximale du détecteur sur l'image : • KLT : valeur propre minimale λ_{min} du tenseur de structure $J(\sigma, D)$ • Harris-Förstner : métrique de Harris $R = \lambda_{max}\lambda_{min} - \alpha(\lambda_{max}+\lambda_{min})^2$ Les points où la réponse est supérieure au seuil sont détectés.	100γ

1. Etude théorique

Rappelez l'influence théorique des différents **hyperparamètres** de ces détecteurs sur les performances de détection.

2. Etude expérimentale

En utilisant les images de test fournies et toutes autres images de votre choix :

2.a Comparer les 2 détecteurs et étudier qualitativement l'influence de leurs hyperparamètres sur leurs performances en termes de fiabilité[1], précision, et robustesse vis-à-vis des propriétés image (**bruit** / **contraste** / **texture** / …).

[1] Dans une perspective d'appariement, *i.e.* caractère distinctif, nombre et distribution spatiale des réponses.

2.b Etudiez expérimentalement la **répétabilité** de ces détecteurs vis-à-vis des conditions d'acquisition suivantes :

- point de vue
- éclairement
- bruit
- sous/sur-échantillonnage (*i.e.* redimensionnement)
- taux de compression

Dans les 4 derniers cas, l'approche adoptée pour générer des échantillons de test consistera à synthétiser des variations d'une image native via des transformations paramétriques[2] pour une gamme de paramètres dont les valeurs seront précisées.

INFORMATIONS COMPLEMENTAIRES

Plusieurs moyens existent afin de sauvegarder les images générées par les notebooks, plutôt que d'effectuer une copie d'écran ou de fenêtre.

1. Si vous disposez d'une installation d'OpenCV compilée avec la bibliothèque Qt, un bouton « Save current image » est disponible sur la barre d'outils supérieure de chaque fenêtre, ainsi que le mnémonique « CTRL + S ».

2. Le fichier `OpenCV_Image_Utilities.py` fournit des fonctions `save_image_as_JPEG()` et `save_image_as_PNG()` permettant de sauvegarder une image avec une qualité maximale, respectivement au format JPEG et PNG.

```
[Numpy array]  image                          # Image à sauvegarder
[Python string]      filename1 = "resultat.jpg"    # Nom du fichier JPEG
[Python string]      filename2 = "resultat.png"    # Nom du fichier PNG

save_image_as_JPEG(image, filename1)
save_image_as_PNG(image, filename2)
```

   Pour sauvegarder les images générées aux différentes étapes de traitement, insérez un appel à l'une de ces fonctions avant chaque instance de `imshow()` dans la fonction `process_display()`.

[2] Pour générer des versions sous/sur-exposées d'une image, appliquez une gamma correction, en utilisant par exemple la routine *imadjust()* de MATLAB.

Pour synthétiser des versions bruitées d'une image, utilisez par exemple la routine *imnoise()* de MATLAB. Précisez la loi et les valeurs des hyperparamètres du(des) modèle(s) de bruit.

La Toolbox Image Processing de MATLAB est disponible sur les serveurs Linux de TSP.