

Application Multimédia

PROJET FINAL

Gaël ANDRIANIRINA, Khanya KOOK BITJOCKA, Grégoire MERCEREAU
Piruntha NADARAJAH, Maxime OUSSOUMANOU, Myriam SAAD MARZOUK et
Julien SCHAMETZ

03 JUIN 2018 | ISEP

Table des matières

Introduction	3
I. Fonctions principales de traitement d'image	4
1. Dilatation/Erosion	4
1.1. Erosion	4
1.2. Dilatation	4
2. Resizing.....	7
3. Lighten / Darken	7
4. Panorama / Stitching	9
5. Canny Edge Detection	10
II. Fonctions avancées : Traitement de vidéo	12
1. Lighten/Darken.....	12
2. Blur	12
3. Gaussian Blur	13
4. Rotation	14
5. Canny Edge Detection	15
III. Interface.....	17
Conclusion.....	20
1. Améliorations envisageables	20
2. Bilan	20

Table des illustrations

Figure 1: Image originale.....	5
Figure 2 : Image érodée.....	6
Figure 3 : Image dilatée.....	6
Figure 4 : Image redimensionnée	7
Figure 5 : Trakbar Lighten/Darken.....	8
Figure 6 : Partie droite de l'image	9
Figure 7 : Partie de gauche de l'image	9
Figure 8 : Panorama : rassemblement des deux blocs précédents	10
Figure 9 : Canny Edge detection (cf Figure 1 pour voir l'originale de l'image)	11
Figure 10 : Trackbar et "Lighten/Darken" sur une vidéo	12
Figure 11: Trackbar et "Blur" sur une vidéo.....	13
Figure 12 : Trackbar et "Gaussian Blur" sur une vidéo.....	14
Figure 13 : Trackbar et rotation d'une vidéo.....	15
Figure 14 : Trackbar et détection de contours dans une vidéo.....	16
Figure 15 : Chemin du fichier demandé à l'utilisateur	17
Figure 16 : Fichier vidéo	17
Figure 17 : Fichier image	17
Figure 18 : Choix traitement d'image.....	18



Introduction

L'objectif du projet est de développer un éditeur d'image comprenant six fonctions principales spécifiques. Le but est de reprendre les méthodes de traitement d'image vues en TP et d'implémenter les fonctions dans un programme final.

Notre groupe est constitué de 7 personnes, et le programme doit être capable d'effectuer les six actions suivantes sur une image donnée : dilatation/érosion, redimensionnement, changement de luminosité, panorama/stitching, et détection de contours.

Afin de réaliser ce projet, nous avons travaillé par petits groupes de 2 à 3 personnes, et nous nous sommes répartis les fonctions à réaliser sur les 2 séances programmées pour la réalisation du projet.

Les technologies que nous avons utilisées sont : Linux, OpenCV et C++.

Afin de faciliter le travail collaboratif, nous avons utilisé Google Drive pour la construction de ce rapport, Messenger pour échanger nos idées et remarques hors du cours , et GitHub afin d'assembler les différentes parties du programme.

I. Fonctions principales de traitement d'image

1. Dilatation/Erosion

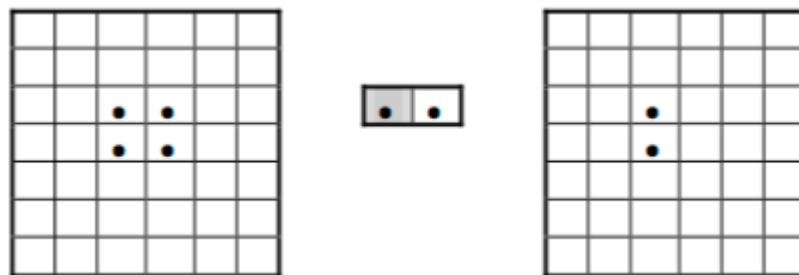
La dilatation et l'érosion sont les deux opérations fondamentales de la morphologie mathématique, à partir desquelles il est possible de construire une multitude d'autres transformations. Elles sont définies à partir de petits ensembles B , appelés éléments structurants pour lequel l'utilisateur fixe la taille. Ceux-ci permettent de caractériser le type de voisinage que l'on souhaite prendre en compte pour traiter l'image ou en extraire des propriétés.

1.1. Erosion

L'érosion d'une image binaire X par un élément structurant binaire B est définie par :

$$E(X,B) = \{y/By \subset X\} = \{y/\forall z \in B, y+z \in X\}$$

L'érodée de X par B est donc l'ensemble des pixels y , tel que l'élément structurant translaté de y est inclus dans X .

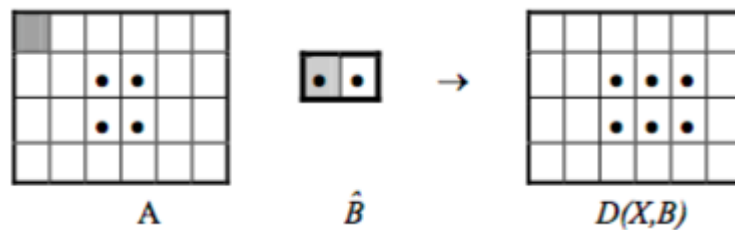


1.2. Dilatation

La dilatation d'un ensemble binaire X par un élément structurant B est définie par :

$$D(X,B) = X \oplus B^* = \{y/y = x + b; x \in X, b \in B^*\}$$

La dilatation revient à parcourir toute l'image. En chaque pixel, il faut placer l'origine de l'élément structurant sur cette position. Si le pixel possède un voisin à 1 au sens de l'élément structurant, alors il est mis à 1, sinon il est mis à 0.



La fonction permet à l'utilisateur de :

- Choisir l'image sur laquelle réaliser l'érosion ou la dilatation
- Choisir entre érosion ou dilatation
- Choisir la taille de l'élément structurant (B)

Voici le résultat du lancement de la fonction avec une taille fixée à 6 pour l'élément structurant :



Figure 1: Image originale



Figure 2 : Image érodée



Figure 3 : Image dilatée

2. Resizing

L'utilisation d'une image nécessite souvent un ajustement de la taille pour l'adapter à un document ou un programme par exemple.

La fonction `resize` permet donc d'agrandir ou de réduire la taille de l'image chargée. Elle est définie par deux paramètres : la largeur et la longueur.

L'utilisateur peut donc faire varier ces deux paramètres. Pour cela, il agit sur les trackbars dédiées : une pour modifier la longueur et une pour modifier la largeur de l'image.

Ces deux chiffres sont les facteurs d'agrandissement, par conséquent s'ils sont compris entre 0 et 1 les proportions de l'image seront réduites et s'ils sont supérieurs à 1 les proportions seront agrandies.

Voici le résultat du traitement de l'image en choisissant l'opération `resize` :

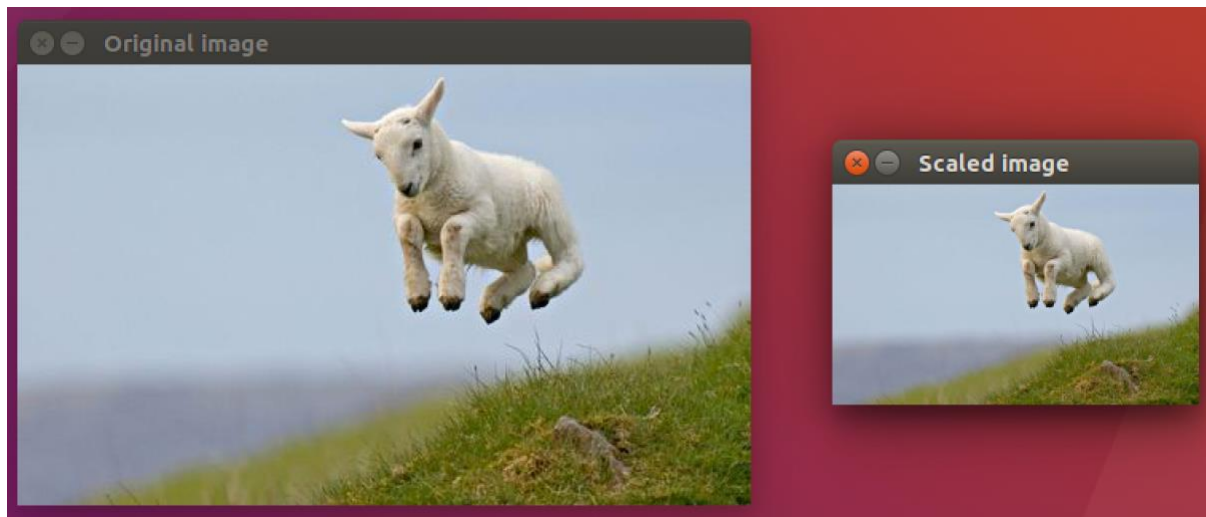


Figure 4 : Image redimensionnée

3. Lighten / Darken

Afin de faciliter la visualisation d'une image et son étude, celle-ci doit avoir une luminosité et un contraste corrects. L'opération d'ajustement de la luminosité est par conséquent une opération commune dans le traitement d'une image.

Cette fonction a donc pour but de permettre à l'utilisateur d'augmenter ou baisser la luminosité de l'image chargée.

Le principe est le suivant : un nombre positif permet d'augmenter la luminosité et un nombre négatif va la réduire. Plus le nombre entré sera grand, plus le changement sera important sur l'image modifiée.

Par exemple, pour augmenter la luminosité de l'image source par un facteur 60. La fonction "lighten" va donc ajouter 60 à chaque pixel de l'image source. Après cela, la fonction va vérifier si les nouvelles valeurs de pixel ne dépassent pas le maximum 255. Dans le cas contraire, c'est-à-dire dans le cas où nous voudrions baisser la luminosité, la fonction va vérifier si les nouvelles valeurs de pixels ne dépassent pas le minimum 0.

12 + 60	23 + 60	84 + 60	122 + 60
123 + 60	34 + 60	92 + 60	200 + 60
23 + 60	45 + 60	29 + 60	73 + 60

=

72	83	144	182
183	94	152	255
83	105	89	133

A l'aide d'une trackbar, l'utilisateur fait varier le paramètre de luminosité.

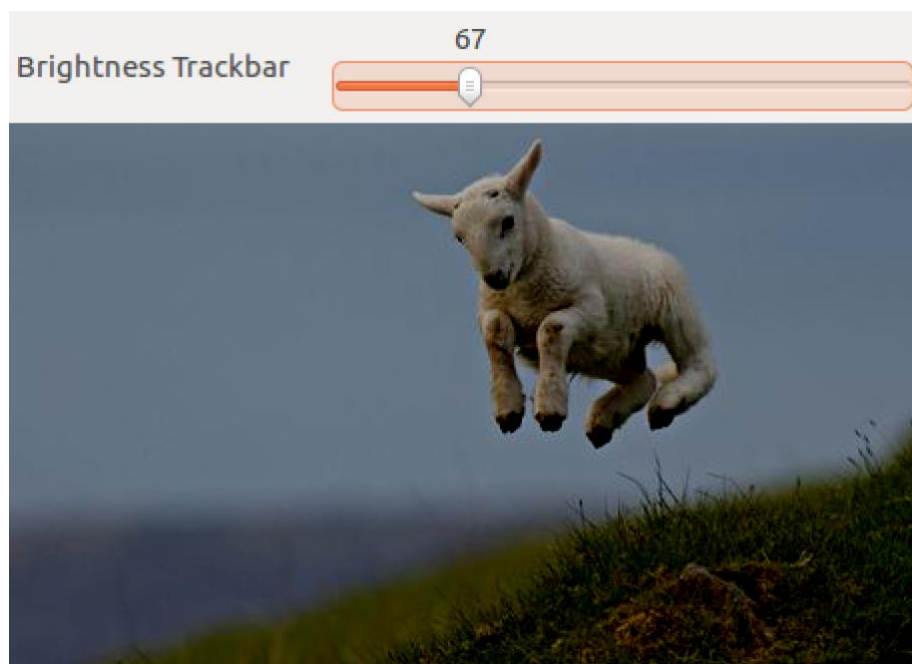


Figure 5 : Trakbar Lighten/Darken

4. Panorama / Stitching

Cette fonction permet de rassembler deux images minimum contenant des blocs adjacents similaires.

L'utilisateur peut entrer deux images qu'il souhaite assembler sous forme de panorama.

Voici le résultat après lancement de la fonction :



Figure 6 : Partie droite de l'image



Figure 7 : Partie de gauche de l'image



Figure 8 : Panorama : rassemblement des deux blocs précédents

5. Canny Edge Detection

La fonction Canny Edge Detection est utilisée pour détecter les contours dans une image. Elle implémente le filtre de Canny, un algorithme conçu par John Canny en 1986 pour être optimal suivant trois critères:

- Bonne détection : faible taux d'erreur dans la signalisation des contours.
- Bonne localisation : minimisation des distances entre les contours détectés et les contours réels.
- Clarté de la réponse : une seule réponse par contour et pas de faux positifs.

La première étape est de réduire le bruit de l'image originale avant d'en détecter les contours. Elle est ensuite lissée à l'aide d'un filtre Gaussien.

Après le filtrage et le lissage, l'étape suivante est d'appliquer un gradient qui retourne l'intensité des contours.

Le résultat obtenu est une carte des gradients d'intensité en chaque point de l'image accompagnée des directions des contours. Par la suite, seuls les points correspondant à des maxima locaux sont considérés comme correspondant à des contours, et sont conservés pour la prochaine étape de la détection c'est-à-dire là où la dérivée s'annule.

Pour finir, la dernière étape est une différenciation des contours par seuillage à hystérésis.

Cela nécessite deux seuils, un haut et un bas; qui seront comparés à l'intensité du gradient de chaque point. Le critère de décision est le suivant. Pour chaque point, si l'intensité de son gradient est :

- Inférieur au seuil bas, le point est rejeté.

- Supérieur au seuil haut, le point est accepté comme formant un contour.
- Entre le seuil bas et le seuil haut, le point est accepté s'il est connecté à un point déjà accepté.

Une fois ceci réalisé, l'image obtenue est binaire avec d'un côté les pixels appartenant aux contours et les autres.

La fonction Canny permet d'effectuer ce traitement sur une image chargée. L'utilisateur définit les deux seuils avec des trackbars.

Résultat de la détection Canny Edge :

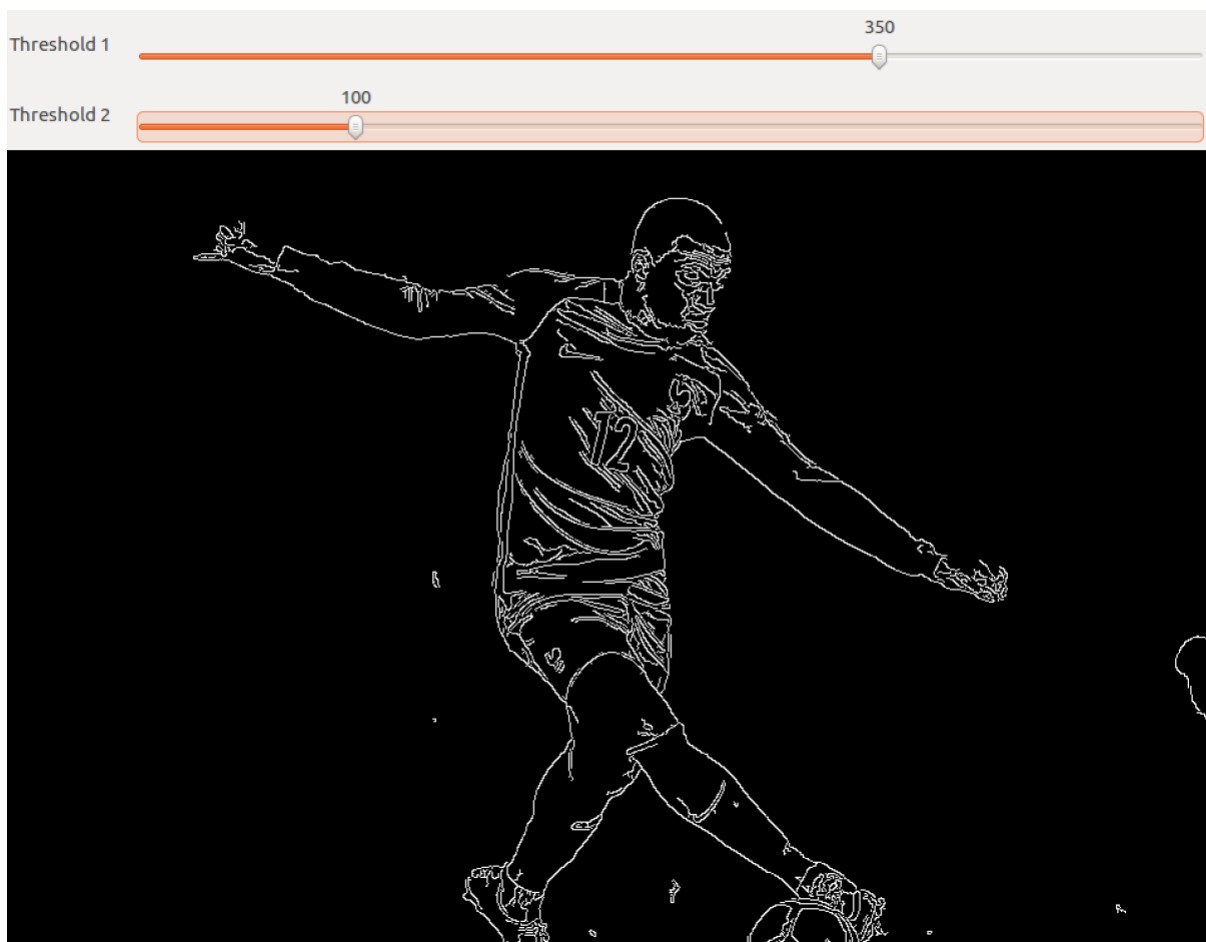


Figure 9 : Canny Edge detection (cf Figure 1 pour voir l'originale de l'image)

II. Fonctions avancées : Traitement de vidéo

1. Lighten/Darken

La fonction « Lighten/Darken » permet de diminuer ou d'augmenter la luminosité de la vidéo. Le même principe appliqué à la vidéo est le même que celui de l'image.

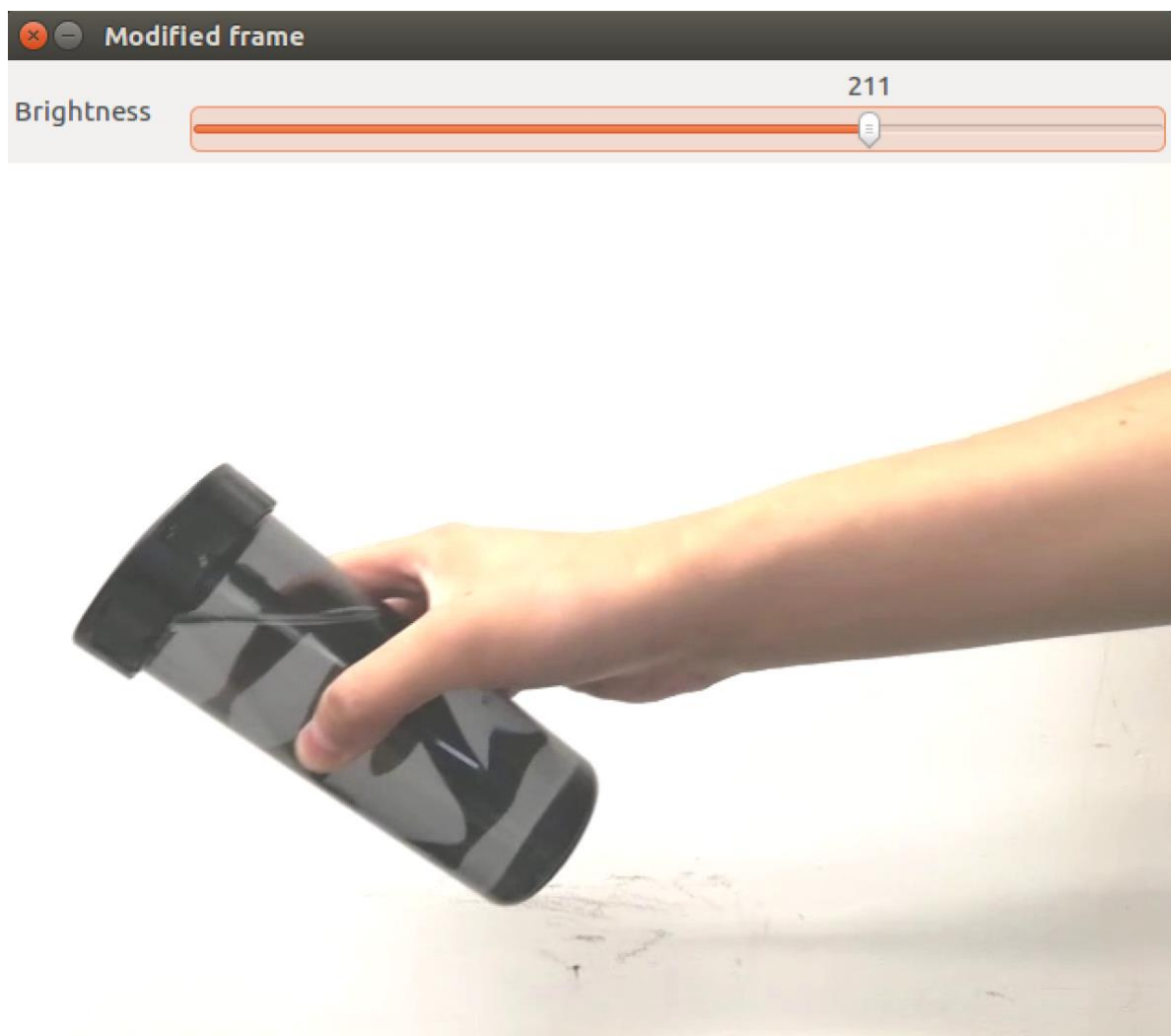


Figure 10 : Trackbar et "Lighten/Darken" sur une vidéo

2. Blur

Cette fonction permet de modifier la netteté de la vidéo. Le niveau de flou est géré par l'utilisateur à l'aide de la trackbar, qu'il peut faire glisser vers la gauche ou vers la droite.

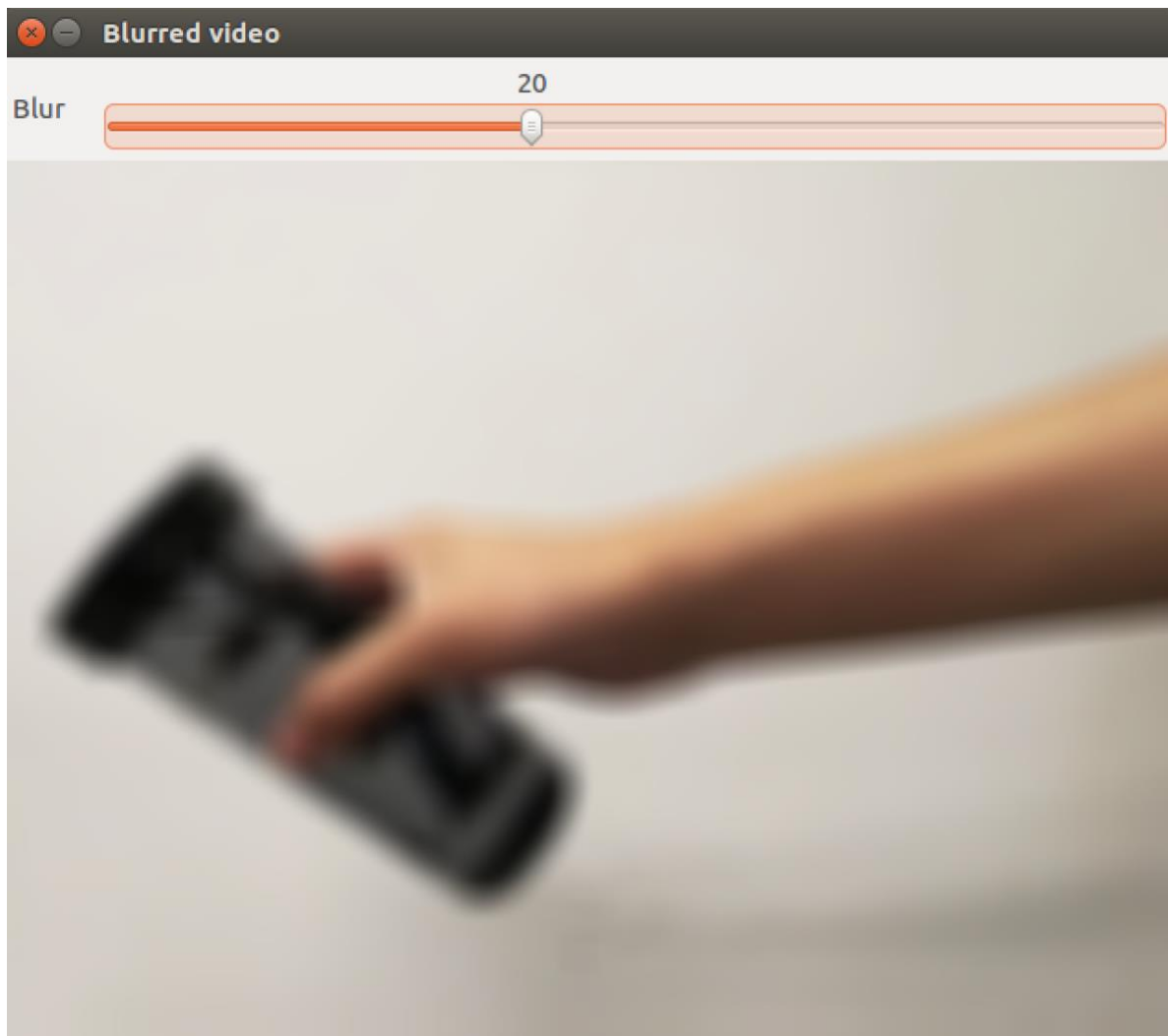


Figure 11: Trackbar et "Blur" sur une vidéo

3. Gaussian Blur

Cette fonction permet d'appliquer un filtre gaussien à la vidéo. Ce filtre joue aussi sur la netteté de la vidéo. Plus la valeur du paramètre est élevée, plus la vidéo est floutée.

Le floutage Gaussien est la méthode de filtrage la plus utilisée. Il permet notamment d'éliminer le bruit dans une image.

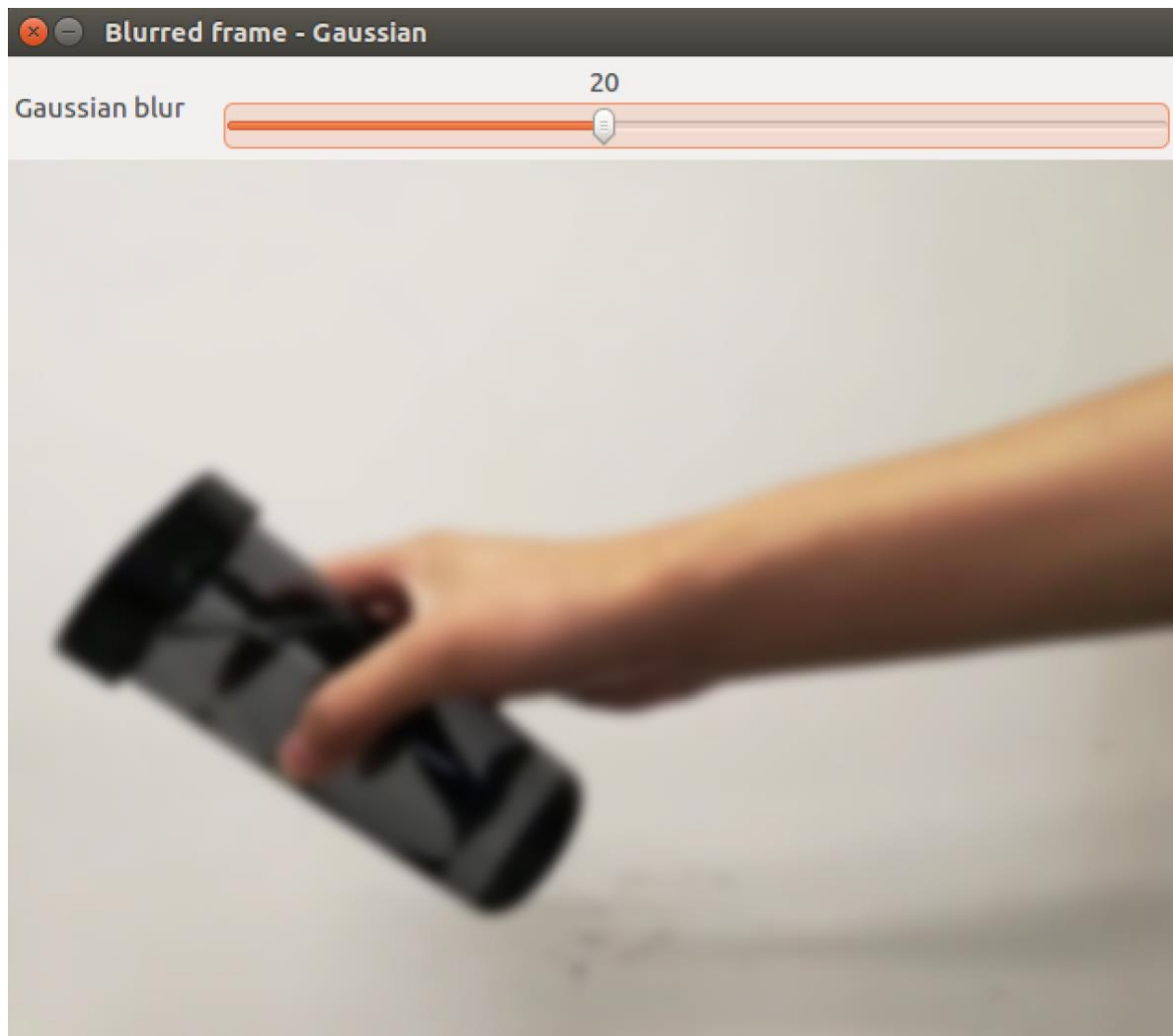


Figure 12 : Trackbar et "Gaussian Blur" sur une vidéo

4. Rotation

Cette fonction permet d'appliquer une rotation à la vidéo. Elle est présente dans la librairie OpenCV.

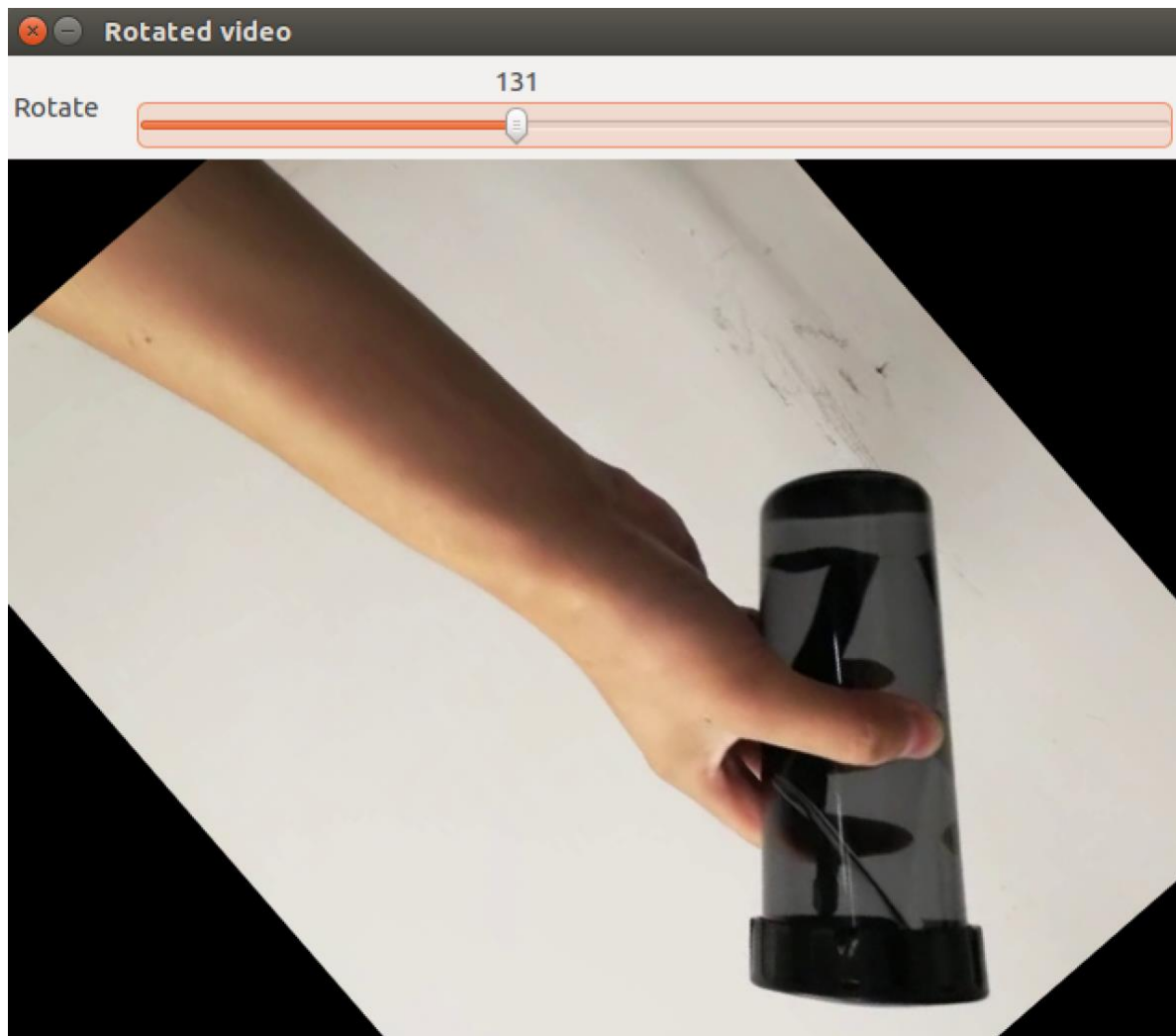


Figure 13 : Trackbar et rotation d'une vidéo

5. Canny Edge Detection

La fonction Canny Edge Detection est la même que pour l'image, mais appliquée à la vidéo. Cette fonction permet la détection des contours dans la vidéo.

La méthode de détection en appliquant la méthode Canny Edge Detection a un faible taux d'erreur. Le résultat possède alors une grande précision. De plus, les contours ne sont marqués qu'une seule fois. Le bruit n'est donc pas détecté par les faux contours.



Figure 14 : Trackbar et détection de contours dans une vidéo

III. Interface

Il est d'abord demandé à l'utilisateur d'entrer le nom de son fichier.

```
Quel est le chemin vers votre fichier ?
```

Figure 15 : Chemin du fichier demandé à l'utilisateur

Si le nom du fichier se termine par “4”, il est considéré qu’il s’agit d’une vidéo et il pourra alors effectuer tous les traitements vidéo existants. Sinon c’est une image et il peut lui appliquer des modifications. .

```
Quel est le chemin vers votre fichier ?
cup.mp4
Que voulez vous faire avec votre vidéo ?
```

Figure 16 : Fichier vidéo

```
Quel est le chemin vers votre fichier ?
foot.jpg
Que voulez-vous faire avec votre image ?
```

Figure 17 : Fichier image

Il doit ensuite choisir quelle modification il souhaite appliquer à son fichier.

Dans le cas du traitement d’une image, les choix suivants sont possibles :

- 1 : dilatation ou érosion
- 2 : fonction redimensionnement
- 3 : fonction permettant de modifier la luminosité de l’image (Lighten/Darken)
- 4 : fonction permettant de faire le panorama
- 5 : fonction permettant de détecter les contours de l’image (Canny Edge Detection).

```

Quel est le chemin vers votre fichier ?
foot.jpg
Que voulez-vous faire avec votre image ?
Dilatation/Erosion = 1
Resizing = 2
Brightness = 3
Panorama = 4
Canny Edge Detection = 5
    
```

Figure 18 : Choix traitement d'image

Et, pour les fonctions de redimensionnement, de modification de luminosité et de détection de contours, l'utilisateur pourra faire varier la valeur des variables en utilisant la trackbar dédiée.

Pour la fonction dilatation ou érosion, l'utilisateur doit entrer les valeurs correspondant au facteurs de dilatation et d'érosion qu'il souhaite appliquer.

Entrer les valeurs de vos parametres :

La fonction panorama demande à l'utilisateur de choisir les deux images qu'ils souhaite associer, et il doit donc taper le nom de celles-ci.

Dans le cas du traitement d'une vidéo, les choix suivants sont possibles :

- 1 : fonction permettant de modifier la luminosité de la vidéo (Lighten/Darken)
- 2 : fonction permettant de flouter
- 3 : application d'un filtre gaussien
- 4 : fonction permettant de faire la rotation de la vidéo
- 5 : fonction permettant de détecter les contours de la vidéo (Canny Edge Detection).

Pour les vidéos, toutes les modifications se font avec une trackbar.

L'utilisateur ne peut effectuer qu'une seule modification à la fois : par exemple, il ne peut pas faire un redimensionnement et un panorama sur une image.

Après l'application de la modification, lorsque l'utilisateur appuie sur la touche escape pour quitter, il lui est demandé s'il souhaite sauvegarder l'image modifiée. Si l'utilisateur tape 1, une nouvelle image avec la modification est sauvegardée.

```
Esc key is pressed by user. Closing the image  
Souhaitez vous enregistrer votre nouvelle image ? Taper 1 pour Oui et 2 pour Non  
1  
Your image has been saved successfully.
```

Conclusion

1. Améliorations envisageables

De nombreuses améliorations sont possibles et envisageables.

Il est notamment possible de permettre à l'utilisateur de pouvoir accoler plus de deux images dans la fonction panorama, car elle est pour l'instant limitée à deux images.

Concernant les fichiers, il faudrait peaufiner la reconnaissance de leur type et ajouter la possibilité de reconnaître d'autres types de fichiers vidéo car le programme actuel ne considère que les noms de fichiers se terminant par « 4 » comme étant des fichiers vidéo.

D'autre part, il serait envisageable d'améliorer l'interface, en ne mettant que des trackbars pour tous les traitements (sur image et vidéo). Par exemple, pour la fonction de dilatation et d'érosion, il serait possible d'ajouter une trackbar pour l'érosion et une trackbar pour la dilatation.

Il serait aussi possible d'afficher toutes les trackbars dans la même fenêtre que le fichier à traiter afin d'améliorer la visibilité de l'utilisateur et simplifier l'utilisation de l'application. Et, lui permettre plusieurs modifications sur ce même fichier.

2. Bilan

Ce projet nous a permis de découvrir et mettre en application les fonctions basiques de traitement d'images pouvant être réalisées avec le langage de programmation C++. Ainsi, nous avons pu mettre à profit les connaissances que nous avons acquises lors des cours et des travaux dirigés d'application multimédia, tout en les croisant avec nos recherches personnelles.

Pour ceux d'entre nous ayant choisi la spécialisation Numérique & Santé, ce projet a permis de faire le lien avec les cours de Vision par Ordinateur. Nous avons pu voir une mise en application concrète des méthodes de traitement d'image étudiées.

Nous pensons avoir globalement réussi à réaliser les fonctions requises lors de ce projet, tout en gardant une optique de travail de groupe en y faisant participer tous les membres.