

Vision Par ordinateur
TP1: **Détection de la peau humaine**
Présenté par **Mongetro GOINT**
Promotion 22 (2017-2019)

=====

Ce rapport est élaboré dans le cadre du TP1 du cours de Vision par ordinateur. Au cours de ce TP, nous avons implémenté un programme permettant de détecter les pixels correspondant à la peau humaine dans une image. Ce document présente de façon générale le fonctionnement, l'implémentation, l'expérimentation et les résultats du programme réalisé.

A- Fonctionnement du programme

Notre programme prend en compte trois arguments à savoir: l'échelle de réduction des valeurs de l'espace des couleurs, la valeur du seuil de décision, puis l'image à évaluer. À la sortie, on a comme résultat: les histogrammes de peau et de non-peau de la base d'apprentissage, l'image d'entrée, l'image obtenue après la détection de la peau, le masque de détection et l'image de référence de peau dans la base d'images.

Pour son fonctionnement, notre programme nécessite une base d'images d'apprentissage. Cette base est représentée par un dossier nommé *base* se trouvant dans le même répertoire que le code source du programme et est constituée à son tour de trois sous-répertoires (*non-peau*, *peau* et *test*) qui contiennent respectivement: les *images non-peau* (images sur lesquelles ne figurent pas de pixels décrivant la peau), les *images peau* (images sur lesquelles apparaissent seulement les pixels décrivant la peau) et les *images de test* (images originales pour obtenir les bases peau et non peau).

➤ **Compilation du programme**

Pour compiler le programme, il suffit de se placer dans le répertoire où se trouvent les codes sources et exécuter, via un terminal, la commande **make**.

Pour exécuter le programme, il faut appliquer les indications données dans la partie fonctionnement du programme, à savoir le nom de l'exécutable suivi des trois arguments représentant l'échelle de réduction des valeurs de l'espace des couleurs, la valeur du seuil de décision et le nom de l'image à évaluer. Exemple : ***./detection_peau_humaine 32 0.4 29.jpg***

B- Implémentation du programme

Dans cette partie nous tenons à présenter le processus d'implémenter de notre programme.

➤ **Création d'une base d'images d'apprentissage**

Comme indiqué dans l'énoncé du TP, nous avons construit une base d'images contenant trois dossiers (*non-peau*, *peau* et *test*). Pour créer cette base nous avons téléchargé des images sur internet et nous les avons choisies en fonction de plusieurs critères comme différence de couleur et différence du type de peau, l'exposition à la lumière et aussi les décors dans les images. Nos sous-répertoires sont constitués de 30 images, chacun. En effet, ce nombre d'images dans la base d'apprentissage est assez grand et diversifié pour permettre à notre programme de produire un résultat satisfaisant.

Afin de pouvoir représenter les parties peau et non-peau respectivement dans les bases *peau* et *non-peau*, il nous a fallu occulter manuellement les pixels peau ou non-peau dans les deux cas en mettant leur valeur à zéro à l'aide du logiciel GIMP.

➤ Construction des histogrammes

Les histogrammes de peau et de non-peau sont très importants dans la détection des pixels représentant la peau. Pour construire ces histogrammes, des prétraitements comme la conversion d'un espace de couleur à un autre ou la diminution des valeurs des pixels sont nécessaires. Ainsi, nous avons converti les images de l'espace de couleur RVB vers l'espace de couleur LAB et que nous avons retenu une valeur d'échelle $32 \times 32 \times 32$ à la place de l'échelle initiale $256 \times 256 \times 256$ (voir énoncé du TP). Dans la pratique, les valeurs de pixels comprises entre deux multiples du facteur « $256/\text{échelle}$ » ont été associées à la valeur du plus petit des multiples considérés. Après la réduction, nous comptons le nombre d'occurrences de chaque couple de valeurs (a , b), dans le répertoire d'images considéré. Au final, nous obtenons un tableau contenant le nombre d'occurrences des différents couples. Ce tableau est donc l'histogramme qui nous servira de regulateur.

Il est à noter que nous ne prenons pas en compte les parties en noire de l'image et la composante L de l'image convertie. L'histogramme obtenu est ensuite lissé afin d'améliorer la qualité de la détection en calculant pour chaque pixel la moyenne des valeurs de ses huit voisins et de sa propre valeur. Enfin vient une étape de normalisation de l'histogramme, consistant à diviser la valeur de chaque case de l'histogramme par la somme des valeurs de toutes les cases. Toutefois, à l'affichage nous avons procédé à l'opération inverse, c'est-à-dire revenir à l'échelle de valeur $256 \times 256 \times 256$.

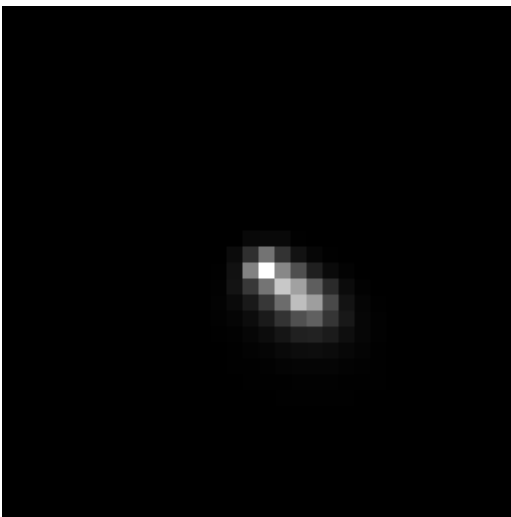


Figure 1 : Histogramme pour peau

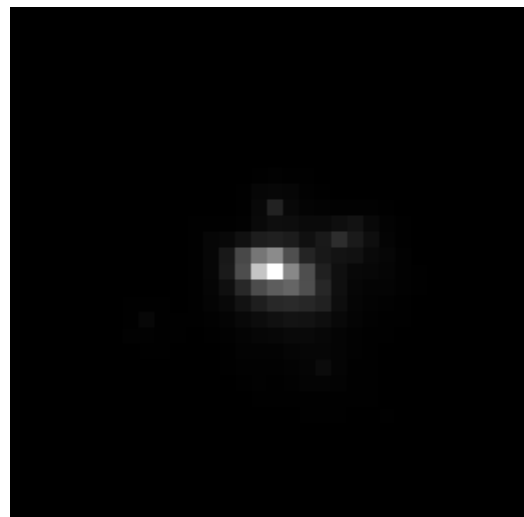


Figure 2 : Histogramme pour non-peau

➤ Détection de la peau dans les images

Dans cette partie, nous allons expérimenter les histogrammes calculés précédemment en identifiant les pixels de peau et ceux de non peau dans des images.

Les manipulations telles la conversion d'un espace de couleur à un autre (RGB vers LAB) et la réduction de la quantification énoncées sont exécutées sur l'image fournie en entrée.

Elle consiste à sélectionner pour chaque couple (a, b) d'un pixel de l'image fournie, les valeurs correspondantes à ce couple dans les histogrammes « peau » et « non-peau ». Nous calculons ensuite la probabilité que le pixel considéré soit un pixel peau en se basant sur la formule de Bayes mentionnée dans l'énoncé du TP. La valeur de la probabilité obtenue est comparée à la valeur du seuil fourni en paramètre au programme. Si la probabilité est supérieure au seuil, le pixel est classé comme « peau ». Dans le cas opposé il est considéré comme « non-peau ». Nous obtenons en fin de compte un masque binaire (partie peau 255, partie non-peau 0) qui sera utilisé par la suite pour l'image de sortie.

Évaluation

Pour évaluer les performances de notre programme sur les images, nous avons implémenté un module d'évaluation en se basant sur la formule ci-dessous :

$Performance = \frac{\#_pixels_corrects}{(\#_pixels_corrects + \#_pixels_faux_pos + \#_pixels_faux_neg)}$ où

$\#_pixels_corrects$ = Nombre pixels correctement détectés.

$\#_pixels_faux_pos$ = Nombre pixels détectés comme peau mais qui sont non-peau.

$\#_pixels_faux_neg$ = Nombre pixels détectés comme non-peau mais qui sont peau.

C- Expérimentations et résultats du programme sur des images

Pour détecter les parties peau et non-peau d'une image, il convient de fournir au programme de bonnes valeurs pour les paramètres à savoir l'échelle et le seuil. En fait, nous avons gardé la valeur d'échelle 32 qui a été indiquée dans l'énoncé du TP et nous avons appliqué sur une même image(29.jpg) les seuils 0.1; 0.3; 0.4; 0.5. Les résultats sont présentés ainsi à travers les images ci-dessous:



Figure 3: image (29.jpg) fournie à l'entrée



Figure 4: résultat pour seuil = 0.1 sur l'image 29.jpg



Figure 5: résultat pour seuil = 0.3 sur l'image 29.jpg

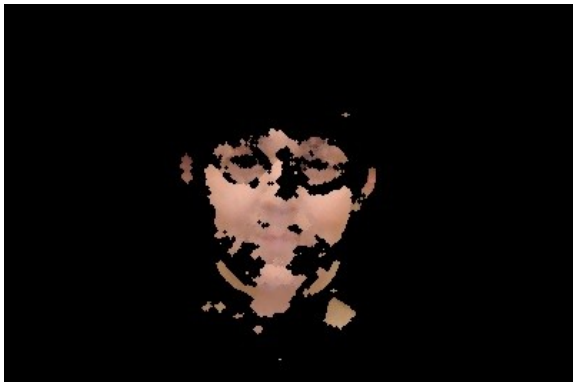
Résultat pour un echelle de 32 et un seuil de 0.1
Performance: 20.08

Résultat pour un echelle de 32 et un seuil de 0.3
Performance: 32.18%



Résultat pour un echelle de **32** et un seuil de **0.4**
Performance: **85.88%**

Figure 6: résultat pour seuil = 0.4 sur l'image 29.jpg



Résultat pour un echelle de **32** et un seuil de **0.5**
Performance: **33.13%**

Figure 7: résultat pour seuil = 0.5 sur l'image 29.jpg

Suite aux expériences ci-dessous avec l'image 29.jpg, nous retenons la valeur du seuil **0.4** et l'échelle **32** comme valeurs applicables pour le reste du travail, car c'est le seuil qui donne le meilleur résultat avec une performance de **85.88%**, et c'est aussi cette echelle qui a été exigé pour ce TP.

➤ Détection de peau dans les images de la base d'apprentissage

➔ *Résultat d'expérience pour la peau de couleur noire*

Echelle = 32; Seuil = 0.4; Nom_image= 2.jpg;



Figure 8: Image de référence



Figure 9: Image de sortie

Les deux images (Figure 8 et Figure 9) montrent que l'exécution de notre programme sur l'image 2.jpg donne un résultat satisfaisant, puisque la majorité des pixels de l'image ont été détectée.

La performance du programme pour ce expérience est de **66.62%**. Toutefois, en depit de cette bonne performance, il est à remarquer que l'image de sortie parrait un peu floue. De plus, les parties correspondants aux yeux et aux sourcils mises à zéro dans notre image de référence (voir *Figure 8*) apparaissent comme des parties peau dans notre image de sortie (voir *Figure 9*). Ceci s'explique par le fait que ces parties dans l'image sont très peu contrastées par rapport au reste des parties peau.

Echelle = 32; Seuil = 0.4; Nom_image= 11.jpg;



Figure 10: Image de référence



Figure 11: Image de sortie

Le résultat de cette expérience sur l'image *11.jpg* montre une performance pas trop considérable, soit **47.26%**. Nous pouvons remarquer que la détection n'est pas optimale et nous obtenons des résultats avec beaucoup plus d'imperfections que dans le cas précédent (voir *Figure 8* et *Figure 9*) lors de la détection de la peau. Des parties peau son considérées comme non-peau (faux négatifs) et vice versa. Cela arrive parce que la peau de l'homme dans l'image était vraiment noire, ce qui crée une certaine similarité entre la valeur des pixels de la couleur des yeux, des sous-cils et d'autres parties par rapport à celle de la couleur de la peau. Mais malgré tout, le programme arrive à détecter des parties peau.

Echelle = 32; Seuil = 0.4; Nom_image= 15.jpg;

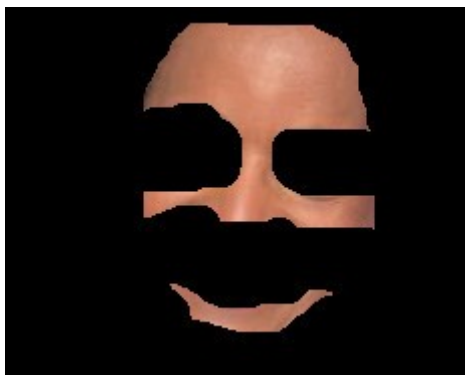


Figure 12: Image de référence



Figure 13: Image de sortie

Pour l'expérience faite sur l'image *15.jpg*, nous obtenons une performance de **42.99%**. Le programme arrive à détecter les parties peau mais moins bien par rapport au cas précédent (voir *Figure 10* et *Figure 11*), car il détecte des parties du contour des cheveux, des vêtements comme des parties peau (voir *Figure 11*), ce qui correspond à des *faux positifs* c'est-à -dire des paties non-peau considérées comme peau. Cependant, nous pouvons constater qu'il arrive à bien détecter les parties correspondant aux yeux et aux sous-cils comme non-peau (voir *Figure 11*). La raison est juste parce que ces parties ont été très contrastées et ont été différentes de celles distinguées comme peau.

→ **Résultat d'expérience pour la peau de couleur blanche**

Echelle = 32; Seuil = 0.4; Nom_image= 6.jpg;



Figure 14: Image de référence

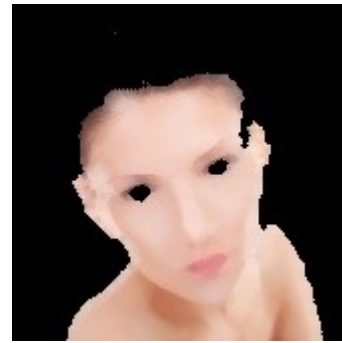


Figure 15: Image de sortie

L'exécution de notre programme sur l'image 6.jpg nous montre des résultats d'une haute performance de **86.14%**. Ce résultat est beaucoup plus considérable que ceux obtenus pour les deux cas précédents (images avec des peaux de couleur noire). Malgré toute cette haute performance, il y a notamment de petites imperfections. D'une part, le programme détecte des parties peau du côté droit du front de la fille comme non-peau (faux négatifs) (voir image 13). D'autre part, il détecte des parties identifiées comme non-peau au niveau des aisselles comme peau (faux positifs). Cependant, le résultat global est assez satisfaisant.

Echelle = 32; Seuil = 0.4; Nom_image= 27.jpg;



Figure 16: Image de référence

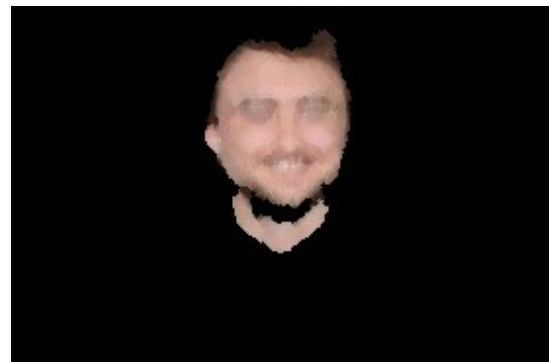


Figure 17: Image de sortie

Dans le cadre de cette expérience sur l'image 27.jpg, nous obtenons encore des résultats avec une performance assez importante, soit **72.48%**. La peau a été bien détectée par le programme et notre image de sortie présente une très grande ressemblance avec l'image de référence (voir image 15). Mais, il est bien de noter qu'il y a des irrégularités, car il considère les parties représentant les yeux, les dents et aussi une partie des cheveux comme peau (faux positifs) (voir Figure 15). Cela se produit parce que l'homme dans l'image avait une partie des cheveux un peu jaunies, des yeux attirés sur maron et des dents presque de la même couleur que la peau de son visage. Donc, cela crée une certaine confusion au programme lors de l'identification de la valeur des pixels représentant peau et non-peau.

Echelle = 32; Seuil = 0.4; Nom_image= 19.jpg;



Figure 18: Image de référence



Figure 19: Image de sortie

Pour cette dernière expérience avec les images de peau de couleur blanche, nous obtenons une performance de **67.34%**. D'abord il est à noter qu'il y a une baisse considérable par rapport au deux autres expériences précédentes faites avec des images de peau de couleur blanche. Ensuite, nous remarquons des imperfections plus poussées au niveau de la détection de la peau. Le programme détecte les parties des yeux, des dents et aussi une parties des cheveux au dessus du front de la dame et aussi du fond de l'image comme peau, donc des (faux positifs) (voir Figure 17). Cette confusion se produit juste parce que la peau de la personne était vraiment blanche et qu'elle avait des cheveux attirés vers gris, des dents blanches et que le fond de l'image était semblable à la couleur de sa peau.

→ Détection dans les images autres que ceux dans la base d'apprentissage

Dans cette partie nous présentons le résultat de nos expérimentations sur des images en dehors de notre base d'apprentissage.



Figure 20: Image de référence

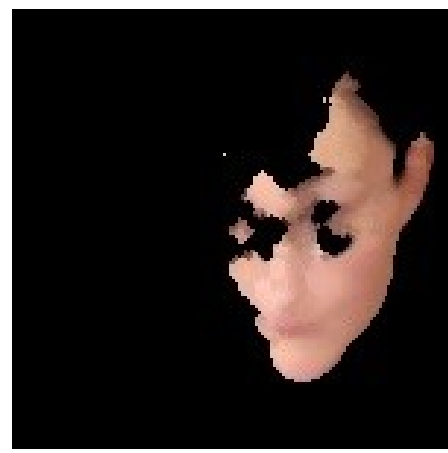


Figure 21: Image de sortie

Performance = **40.91%**



Figure 22: Image de référence



Figure 23: Image de sortie

Performance = 62.18%

Pour les expérimentations faites avec des images hors de la base d'apprentissage, nous remarquons que la détection de la peau s'effectue aussi, mais le programme fonctionne un peu moins bien par rapport aux résultats obtenus avec les images de la base. Nous constatons surtout qu'il détecte des parties peau comme non-peau.

Conclusion

Somme toute, après nos expérimentations avec le programme implémenté, nous avons conclu que le détecteur fonctionne bien car il arrive à détecter la peau quelque soit la couleur. Mais il y a, entre autres, des conclusions à porter en plus. D'abord, dans certains cas exceptionnels, il y a des irrégularités de détection puisque le programme détecte parfois des éléments n'étant pas des pixels non-peau en tant que peau (faux positifs) et vice-versa. Ensuite, il est à noter que la détection est plus optimale sur des images avec la peau blanche plutôt que sur des images avec la peau noire. En plus, la détection est moins performante quand il y a des parties contenant des valeurs de pixels non-peau qui ressemblent à des parties peau.

Enfin, la performance de notre programme peut encore être améliorée, premièrement en choisissant des images où les parties représentant la peau sont vraiment différentes des parties non-peau; deuxièmement en augmentant le nombre d'images dans la base d'apprentissage.

Références bibliographiques

1- http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html#linux-installation

2- ezhnevets V., Sazonov V., Andreeva A., "A Survey on Pixel-Based Skin Color Detection Techniques". Proc. Graphicon-2003, pp. 85-92, Moscow, Russia, September 2003.

<http://graphics.cs.msu.ru/en/publications/text/gc2003vsa.pdf>