

VISION PAR ORDINATEUR TP1 :DETECTION DE PEAU DANS IMAGES

Réalisé par : DAOUDA KADRI SAIDI

1	Contexte et objectif	2
2	Description du fonctionnement	2
3	Description de l'implémentation.....	2
3.1	Construction de la base.....	2
3.2	Construction de l'histogramme	3
3.3	Détection de peau dans l'image.....	3
3.4	Evaluation.....	4
3.5	Expérimentations Résultats et Analyse	4
3.5.1	Effet du seuil sur la détection de la peau.....	4
3.5.2	Détection des images de la base d'apprentissage.....	6
3.5.3	Détection dans des images hors de la base d'apprentissage	7
4	Conclusion	8

TABLE OF FIGURES

Figure 1	Histogramme Peau.....	3
Figure 2	Histogramme non-peau.....	3
Figure 3	image de référence.....	6
Figure 4	Image détectée	6
Figure 5	Image de référence.....	6
Figure 6	Image détectée	6
Figure 7	Image de référence.....	7
Figure 8	Image détectée	7
Figure 9	Image détectée	7
Figure 10	Image de référence.....	7

1 Contexte et objectif

Les images sont des données qui cachent plusieurs informations importants, Début l'aube l'homme utilise l'image pour communiquer et transmettre le savoir. L'analyse d'image présente un intérêt majeur dans tous les domaines où le problème de la recherche automatique d'informations dans des images est incontestable.

L'objectif de ce TP est d'implémenter un programme robuste de détection de peau humain dans une image. L'idée est simple, il s'agit d'identifier les pixels de la peau quelque choix la luminosité ou la couleur dans les images qui seront passés au programme.

2 Description du fonctionnement

Le programme mise en place prend certains arguments: l'échelle de réduction de l'espace de couleur, la valeur de seuil de détection, et l'image à traiter. Nous avons comme sortie du programme les histogrammes de peau et de non-peau de la base d'apprentissage, l'image entrée, l'image obtenue après détection de la peau, le masque de détection et l'image de référence correspondante dans la base d'images de peau.

Afin de lancer le programme l'utilisateur doit effectuer les tâches ci-dessous:

- **Composition de la base de données:** Dans le même répertoire du programme se trouver un dossier nommé **<db_image>** qui contient trois répertoires **mask_peau** (iniquement les pixels qui identifier la peau humain), **mask_nonpeau** (le contraire de mask_peau) et enfin **originale** (image à travers lequel nous avons défini le mask peau et non peau).
- **Compilation du programme:** Pour compiler, il faut d'abord se placer dans le répertoire contenant le code source et puis exécuter la commande **make** à travers le terminal.
- **Exécution du programme:** il suffit de lancer la commande **./detection_peau <echelle> <seuil> <nom_image_test>**.
 Echelle : signifier le degré de réduction de l'espace couleur.
 Seuil : Valeur qui permet d'identifier les pixels peau et autre
 nom_image_test : nom de l'image fournie en entrée au programme

3 Description de l'implémentation

3.1 Construction de la base

Notre base d'images est composée de trente-six (36) images obtenues auprès de chaque étudiant de la promotion 22. Ces images, contiennent des personnes de différentes couleurs, habillées différemment et dans des décors divers. Nous avons aussi veillé à ce que les expositions aux sources lumineuses soient aussi différentes que possible afin d'assurer une base d'apprentissage suffisamment diversifiée à notre programme. Une fois les images sélectionnées, il s'agit dans un premier temps de créer un groupe d'images qui ne font ressortir que les pixels décrivent la peau des individus (la valeur des autres pixels étant mise à 0) et dans un second temps d'effectuer l'opération inverse c'est à dire créer des images dans lesquelles seule la peau était occultée (valeur des pixels mise à 0). Les images montrant la peau ont été regroupées dans un répertoire nommé « mask_peau » et celles l'occultant

dans un répertoire nommé « mask_non-peau ». Les images originales ayant servi à la création des deux groupes précédemment décrits ont quant à elles été conservées dans un répertoire nommé « test ». Dans l'étape suivante (celle de la construction des histogrammes), seules les images des répertoires « peau » et « non-peau » qui seront exploitées.

3.2 Construction de l'histogramme

L'une des étapes très importants dans le processus de détection des pixels de peau est la construction des histogrammes **peau et non peau**. Dans chaque cas, il était question de convertir l'espace de couleurs RGB à l'espace de couleur Lab. Ensuite, dans l'optique de réduire le temps de calculs à effectuer, nous avons réduit l'espace des valeurs des pixels de $256*256*256$ à échelle où « échelle » est la première valeur fournie en paramètre au programme.

Par exemple si l'échelle fournie est 32, nous réduirons l'espace des valeurs de $256*256*256$ à $32*32*32$. Pour ce faire nous associons toutes les valeurs de pixels comprises entre deux multiples du facteur « $256/\text{échelle}$ » à la valeur du plus petit des multiples considérés. A titre exemple, pour une échelle de réduction égale à 32, toutes les valeurs comprises entre 0 et 8 (exclu) seront associées à la valeur 0 dans le nouvel espace.

Après la réduction, nous comptons le nombre d'occurrences de chaque couple de valeurs (a , b), dans le répertoire d'images considéré. Ainsi à la fin de cette étape, nous obtenons un tableau contenant le nombre d'occurrences des différents couples. Ce tableau n'est rien d'autre que l'histogramme. 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. Une fois l'histogramme construit, nous lui appliquons un lissage afin d'améliorer la qualité de la détection. Rappelons que le lissage consiste à calculer pour chaque pixel la moyenne des valeurs de ses huit (08) voisins et de sa propre valeur. Nous finalisons cette étape en normalisant les valeurs de notre histogramme. Ladite normalisation consiste à diviser la valeur de chaque case de l'histogramme par la somme des valeurs de toutes les case de l'histogramme.

Notons que pour l'affichage de l'histogramme nous remettons l'histogramme à l'échelle $256*256*256$ pour une meilleure visualisation. L'opération est exactement l'inverse de celle ayant permit la réduction de l'espace des valeurs. Les figures 1 et 2 montre les histogrammes « peau » et « non-peau » obtenus à partir de notre base d'images.

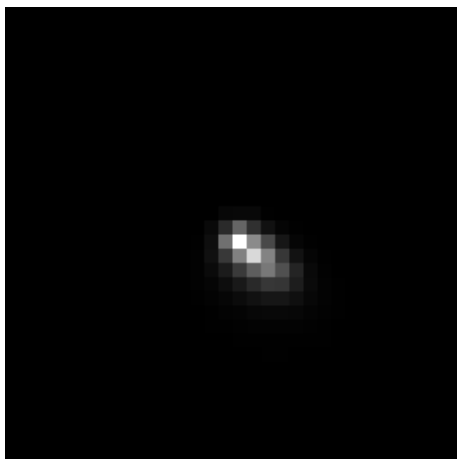


Figure 1 Histogramme Peau

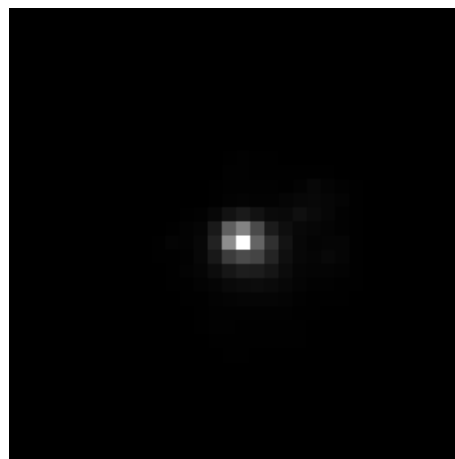


Figure 2 Histogramme non-peau

3.3 Détection de peau dans l'image

Dans cette partie, nous allons nous baser sur les histogrammes déjà construits afin de détecter les pixels peau dans les images qui seront passées au programme. La procédure est la suivante : En

premier étape, nous chargeons l'images dans l'espace RGB puis nous convertissons en Lab. Ensuite la seconde étape consiste à réduire la taille des pixels et enfin la détection proprement dite.

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 ». Puis à calculer la probabilité que le pixel considéré soit un pixel peau en se basant sur la formule de Bayes décrite dans l'énoncé du TP. Par la suite la probabilité obtenue est comparée à la valeur de seuil fournie 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 ». Il est alors construit un masque où les pixels peau ont la valeur 255 et ceux non-peau la valeur 0. Ce masque sera, pour finir, utilisé pour restituer l'image détectée.

3.4 Evaluation

Nous avons proposé un module d'évaluation afin de connaître les performances de notre programme. La performance nous est donnée par la formule utilisée dans le TP3 du cours de traitement d'image.

$$\text{Performance} = \frac{\text{nb_pixels_corrects}}{(\text{nb_pixels_corrects} + \text{nb_pixels_faux_positifs} + \text{nb_pixels_faux_negatifs})}$$

nb_pixels_corrects : Nombre pixels correctement détectés

nb_pixels_faux_positifs : Nombre pixels détectés comme peau mais qui sont non-peau

nb_pixels_faux_negatifs : Nombre pixels détectés comme non-peau mais qui sont peau

Notons toutefois que cette mesure est entachée d'un certain biais vu que nous n'avons pas pu établir des vérités terrain parfaites c'est à dire des images de référence (images « peau ») parfaites. Nous l'utilisons donc à titre indicatif seulement.

3.5 Expérimentations Résultats et Analyse

Dans cette partie est consacrée à la présentation de nos expérimentations, résultats et analyses. Nos expérimentations ont été réalisés sur deux vols à savoir tests sur les images de la bases d'apprentissage, tests sur les images qui fait partie de la base.

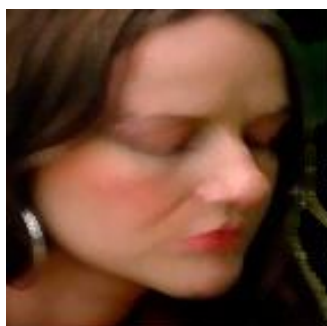
Afin de trouver la meilleure valeur de seuil à appliquer dans nos expériences, une série de tests s'avère nécessaire. Les résultats sont affichés ci-dessous:

3.5.1 Effet du seuil sur la détection de la peau

Les images ci-dessous ont été obtenues en appliquant différents seuils ($0,1 < \text{seuil} < 0,7$) de détection à l'image test 22.jpg.



Seuil: 0 ; perf: 49,47 %



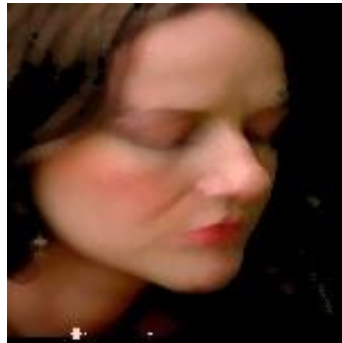
Seuil: 0,1 ; perf: 50,83 %



Seuil: 0,3 ; perf: 53,27 %



Seuil: 0,2 ; perf: 51,22 %



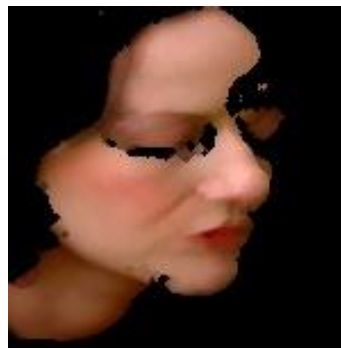
Seuil: 0,4 ; perf: 67,89 %



Seuil: 0,5 ; perf: 69,67 %



Seuil: 0,6 ; perf: 69,57 %



Seuil: 0,7 ; perf: 72,20 %

A travers les tests réalisés, nous constatons que la valeur du seuil = 0,7 donne une meilleure performance. Avant de continuer notre expérience nous allons vérifier de la valeur de seuil sur d'autre image.



Seuil:0,7 ; perf= 22,73 %



Seuil:0,4 ; perf= 57,09 %

A partir des performances ci-dessus, nous avons jugé utile utilisé le seuil 0,4 et l'échelle 32 dans la suite de nos expériences.

3.5.2 Détection des images de la base d'apprentissage

./detection 32 0,4 2.jpg performance = 67,76%



Figure 3 image de référence



Figure 4 Image détectée

La performance obtenue pour cette image est très bonne car il détecte parfaitement des pixels « peau ». On remarque donc que le programme détecte bien la peau de couleur noir.

./detection 32 0,4 8.jpg performance = 85,56%



Figure 5 Image de référence

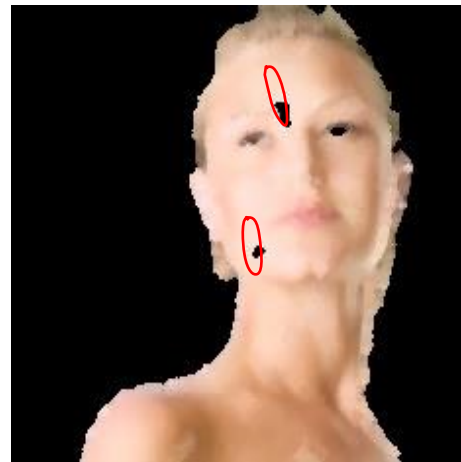


Figure 6 Image détectée

Comme pour l'image précédente, la peau est bien détectée même si quelque faux négatifs (encadrés en rouge). Ajoutons qu'ici la peau détectée est de couleur blanche ce qui signifie que le programme a bien appris les caractéristiques de différents types de peaux.

En conclusion de nos expérimentations sur des images de la base d'apprentissage, nous pouvons retenir que la peau est bien détectée.

3.5.3 Détection dans des images hors de la base d'apprentissage

/detection 32 0,4 base1.jpg performance = 72,14%



Figure 7 Image de référence

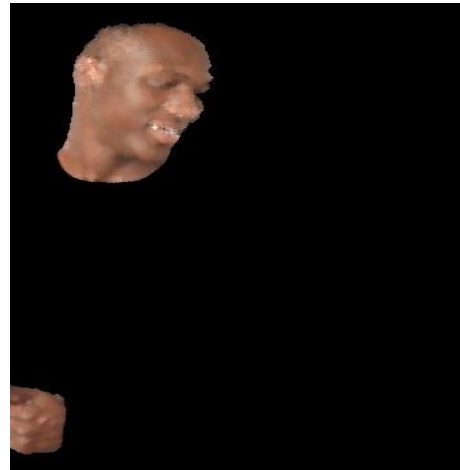


Figure 8 Image détectée

/detection 32 0,4 35.jpg performance = 85,65%



Figure 9 Image détectée

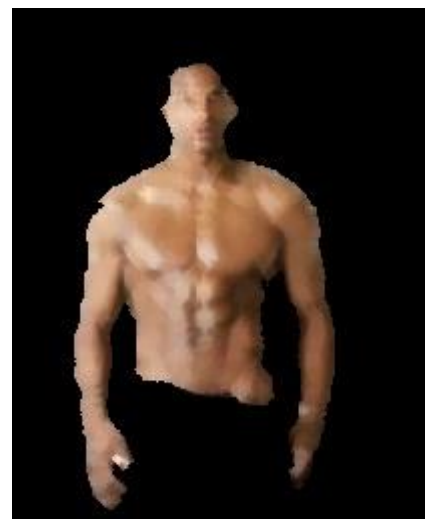


Figure 10 Image de référence

Sur les différentes images testées ont notes que la peau est bien détectée.

4 Conclusion

Dans ce TP, nous avons pu mettre en place un programme de détection de peau dans des images. Ainsi, nous avons utilisé une ensemble de 30 images aussi diverses que possible pour créer des images ne faisant ressortir que la peau afin de mettre en place notre base d'apprentissage. Après ces expérimentations nous pouvons conclure que la détection de la peau a été bien réalisé quel que soit la couleur même si dans certains cas des éléments n'étant pas des pixels peau ont été détectés comme tel ou que des parties de la peau sont considérées comme non-peau. Ainsi, ce travail sur la détection de la peau dans une image nous a profondément permis de comprendre l'utilité et le mécanisme qui l'accompagne.