

Matière : Vison par Ordinateur

TP1 :Détection de la peau par les techniques de vision par Ordinateur

Auteur : Oumarou Altine Mohamadou Aminou

1 Introduction

Le présent rapport fait la description du TP1 réalisé dans le cadre du module de Vision par Ordinateur. Notre objectif ici est de pouvoir implémenté un programme capable de détecté une peau d'image dans une base d'image peau et non-peau ainsi que l'espace de couleur **LAB**.Ce rapport a pour but donc d'expliquer les différents travaux réalisé ,les analyses ainsi que les commentaires et les différentes améliorations possible.

QUELLE EST LA DÉMARCHE A SUIVRE ?

De prime abord pour atteindre cet objectif nous allons nous baser sur l'indice couleur qui est la primitive la plus simple à calculer et la plus riche. Dans un premier temps nous allons utiliser l'espace **LAB** (uniquement A et B) pour représenter les histogrammes de pixels de peau et de non-peau. Subsidiatement, nous allons établir les probabilités pour qu'un pixel soit un pixel de peau dans chaque combinaison d'axes. Et enfin la phase de segmentation qui nous permettra d'avoir des régions de peau cohérentes souhaités.

2 Fonctionnement

Le programme que nous avons implémenté prend en compte comme souhaité dans le TP : l'image d'entrée (image à évaluer), l'échelle de réduction des valeurs de l'espace de couleurs, et la valeur seuil de décision. Nous obtenons en sortie les deux Histogrammes (peau et non-peau), l'image entrée, l'image obtenue après détection de la peau, l'image de référence correspondante dans la base d'images de peau et le masque de détection.

COMMENT EXÉCUTE-T-ON LE PROGRAMME ?

Pour tester notre programme nous avons mis en place une base d'image contenant trois (3) catégories (répertoires) d'images à savoir répertoire peau qui contient uniquement les images dont les pixels identifie la peau, répertoire non-peau qui contient uniquement les images ou ne figure pas les pixels peau, et le répertoire test qui contient les images originales qui ont été traités pour obtenir les deux répertoires précédent.*Il est également a noté que chacun des différents répertoire contient au plus 30 images.* Alors ce fais pour excuter le programme

- on se place dans le dossier contient le programme en question avec la commande **cd**
- puis exécuter la commande **make** afin de pouvoir compiler.
- Ensuite exécuter le programme au moyen de la commande : **./skin-detection " Echelle-de-reduction " " seuil " " nom-mage-test "**

Cette commande d'exécution spécifié ce que nous avons révélé plus-haut ou **Echelle-de-reduction** correspond à l'échelle de réduction des valeurs de l'espace de couleur, seuil qui caractérise la valeur de seuil de décision et **nom-mage-test** nom de l'image d'entrée.

3 Description de la phase d'implémentation

Pour la phase d'implémentation du programme nous avons subdivisé ce dernier en quelques parties essentielles :

1. Création de la base images
2. Constructions des Histogrammes
3. Processus de détection de la peau dans une image
4. Le Processus du Post-traitement

3.1 Création de la base images

Dans l'optique de pouvoir constituer la base d'image, nous avons collecté de part et d'autre les images sur le net en vue de la création de la base notamment sur **google image** que nous avons traité en vue de pouvoir spécifier les images peau et non-peau. En effet ces images sont formées de plusieurs personnes différentes avec chacune une couleur bien de peau bien distincte. Et cela a été fait en vue de respecter les consignes données par le TP dans le but d'avoir des bons résultats. Pour le traitement de la base, nous avons procédé de la manière suivante :

- pour la création de la base peau nous avons mis tous les autres pixels n'ayant aucun trait à la peau à zéro (0);
 - pour la base non-peau c'est l'opération contraire qui a été effectuée c'est à dire tous les pixels peau ont été mis à zéro (0);
 - pour les images ayant servi aux opérations précédentes ils ont été classés dans la base test.
- Après cette phase nous passons à l'étape pour la construction de l'histogramme.

3.2 Constructions des Histogrammes

Dans ce cas de figure à noter que nous allons seulement utiliser la base image peau et non-peau. Comme recommandé dans le TP nous allons construire deux Histogrammes pour cette partie l'un pour les images de la peau et l'autre pour ceux de non-peau. Pour ce faire nous allons convertir au moyen de la fonction "CV_BGR2Lab" d'OpenCV l'ensemble des images de l'espace RGB vers l'espace LAB qui nous a été imposé. Par la suite, dans un souci de diminution des calculs à effectuer, nous avons réduit l'espace des valeurs des pixels de **256*256*256** à **échelle*échelle*échelle** comme il a été recommandé dans l'énoncé du TP où « échelle » est la première valeur fournie en paramètre au programme. Par exemple si l'échelle fournie est 35, nous réduirons l'espace des valeurs de **256*256*256** à **35*35*35**. Pour ce faire nous associons toutes les valeurs de pixels comprises entre deux multiples du facteur « **256/échelle** » à la valeur du plus petit des multiples considérés. A titre d'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. Tous ces étapes que nous venons d'effectuer sont des recommandations de l'énoncé du TP, cependant nous avons ajouté certaines de nos idées mais nos plus gros apports sont un peu plus loin. Après la réduction, nous passons à la construction de l'histogramme en tant que tel pour ce faire nous comptons le nombre d'occurrences de chaque couple de valeurs (a, b), dans le répertoire d'images considéré qui va nous permettre d'obtenir un tableau contenant le nombre d'occurrences des différents couples : **c'est l'histogramme**.

Dans cette partie nous ne prenons pas en compte les parties de l'image qui sont à zéro (0), aussi nous ne prenons pas en compte la composante L de l'espace LAB qui ne nous intéresse pas. Ainsi dans le souci d'améliorer la qualité du résultat nous appliquons le lissage à l'histogramme. Cette dernière consiste à calculer la moyenne des huit (8) voisins + la valeur elle-même pour chaque pixel. Enfin nous finissons la construction de ces histogrammes en les normalisant. Cette opération consiste à

diviser la valeur de chaque case de l'histogramme par la somme des valeurs de toutes les cases de l'histogramme. Ci-dessous les figures qui montrent l'histogramme de l'image peau et non-peau du modèle qui a été obtenue à partir de notre base d'image créée.

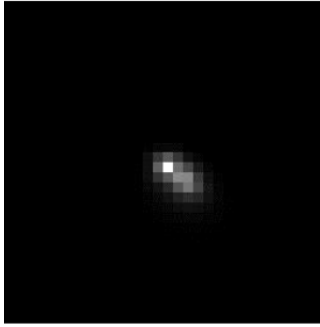


FIGURE 1 – Histogramme peau

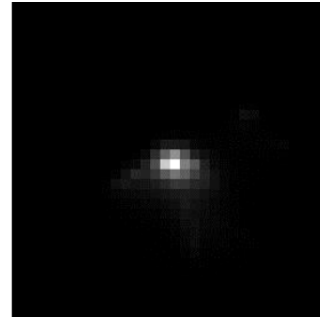


FIGURE 2 – Histogramme non-peau

3.3 Processus de détection de la peau dans une image

Dans cette phase nous allons exploiter les histogrammes construits à l'étape précédente pour détecter les pixels peau des images fournies au programme. Ainsi pour ce faire nous chargeons dans un premier temps l'image fournie en paramètre dans l'espace RGB puis nous la convertissons dans l'espace LAB au moyen de la fonction "CV_BGR2Lab" d'opencv, puis nous réduisons les valeurs de pixel et enfin nous procédons à la détection de la peau. La technique 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", ensuite calculer la probabilité que le pixel considéré soit un pixel peau ou non en se basant sur la formule de Bayes : Ensuite le résultat obtenu est comparé à la

$$p(\text{peau} \mid a,b) = \frac{p(c \mid \text{peau}) p(\text{peau})}{p(c \mid \text{peau}) p(\text{peau}) + p(c \mid \neg \text{peau}) p(\neg \text{peau})}$$

valeur de seuil passé en paramètre. Si la probabilité est supérieure au seuil alors le pixel est classé comme "peau", dans le cas contraire il est classé comme "non-peau". Ainsi avec ce résultat nous construisons un masque où les pixels peau ont les valeurs 255 et ceux non-peau la valeur 0. Ensuite ce masque est utilisé pour restituer l'image détectée.

3.4 Le Processus du Post-traitement

Cette partie va en effet nous permettre d'améliorer certains résultats de la détection de peau obtenue. En effet de nombreuses déformations sont constatées dans le processus de détection de la peau, ainsi pour pallier certains de ces problèmes nous allons utiliser les opérations de morphologie vue au cours du module de traitement d'image. A cet effet la **morphologie** mathématique ne s'appuie pas sur le traitement du signal, mais repose sur la théorie des ensembles, ce qui en fait une discipline relativement « auto-contenue » et formant un tout cohérent.

Précisons : il est à noter que ce traitement est effectué sur toutes les images résultats (image de sortie).

3.4.1 Évaluations

Dans cette partie nous proposons un petit modèle pour évaluer la performance de notre programme. Nous nous sommes inspirés de ces articles [1][2] pour le mettre en œuvre. Il consiste à calculer la performance du programme en fonction du nombre de pixel bien classifié (pixels correctement détectés), du nombre de pixel détecté comme peau mais qui ne le sont pas (faux positif) et fonction du nombre de pixel détecté comme non-peau mais qui sont peau (faux négatif). La formule est la suivante : avec

$$Performance = \frac{PC}{PC + PFN + PFP}$$

- **PFN** : Nombre de Pixel détecté comme étant non-peau mais qui sont peau ;
- **PC** : Nombre de Pixel correcte ;
- **PFP** : Nombre de Pixel détecté comme étant peau mais qui ne le sont pas.

3.5 Expérimentation et Résultat

Dans cette partie nous allons présenter nos expérimentations, les résultats qui en découlent et les commentaires. Ainsi pour mieux appréhender notre travail nous avons subdivisé cette partie en trois (3) sous parties :

1. Test sur les images de notre base d'apprentissage,
2. Test sur les images hors base ,
3. et le Post-traitement pour améliorer certains défauts lors du processus de la détection de la peau.

3.5.1 Recherche du Seuil optimal

Pour choisir un seuil optimum pour l'expérimentation nous effectuons une série de test avec une même image avec des valeurs de seuil différentes. Les images ci-dessous présentent les résultats :



FIGURE 3 – seuil = 0.1, perf = 65.91 %



FIGURE 4 – seuil = 0.2, perf = 75.65 %

Nous constatons d'après les résultats obtenus ci-dessus que nous avons un bon pourcentage avec les seuils (**0.3 et 0.2**) testés. Donc c'est qui nous amène à déterminer le seuil le plus optimale parmi ces deux(2) derniers. Les images ci-dessous montrent les résultats des deux seuils (0.2 et 0.3) avec une



FIGURE 5 – seuil = 0.3, perf = 81.076 %



FIGURE 6 – seuil = 0.4, perf = 70.71 %

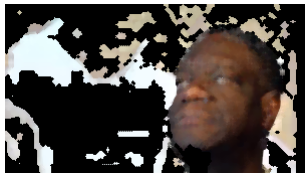


FIGURE 7 – seuil = 0.2, perf = 40.56 %



FIGURE 8 – seuil = 0.3, perf = 55.576 %

nouvelle image :

Au vu de ces résultats nous remarquons que le seuil 0.3 est meilleur. De ce fait nous allons réaliser nos tests avec ce seuil et d'autres seuils pour comparaisons. Mais il est à noter aussi que ce seuil n'est pas optimal pour certaines images, nous verrons ce cas un peu plus loin.

3.5.2 Expérimentation avec les différentes images de la base d'apprentissage

Pour cette partie nous allons considérer l'image du nom «12.png». L'exécution se fait à l'aide de la commande :

— (*./skin-detection 32 0.3 12.png perf:71.47%*)



FIGURE 9 – img sortie



FIGURE 10 – img référence

A travers ces résultats nous pouvons remarquer que nous obtenons une très bonne performance, ce qui veut dire que notre programme détecte en grande partie les pixels peau. Cependant nous pouvons aussi remarquer certaines anomalies : la présence de faux négatif référencé par la couleur rouge. Nous remarquons aussi la détection des cheveux comme peau cela s'explique par le fait que dans l'image de référence les cils des yeux qui ont la même couleur que les cheveux n'ont pas été supprimés du coup le programme considère cette partie comme étant peau. Avec ces résultats obtenus nous pouvons dire que notre programme s'adapte au changement de couleur de la peau vu que la différence de la couleur de la peau sur le visage n'a pas impacté la performance du programme.

— (*Sur image 20.jpg ./skin-detection 32 0.3 20.jpg perf:87.339%*)

De par ces résultats nous remarquons que le programme détecte bien les pixels peau avec une performance de 87.339% ce qui est un bon taux, tout de même nous remarquons la présence d'un faux positif (les yeux) qui peut s'expliquer par le fait que le programme ne sait pas ou classer ces pixels vu qu'ils ne sont ni catégorisés dans la base peau ni dans non peau.

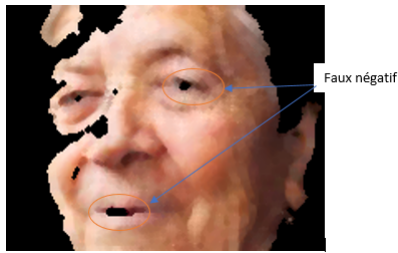


FIGURE 11 – img sortie



FIGURE 12 – img référence

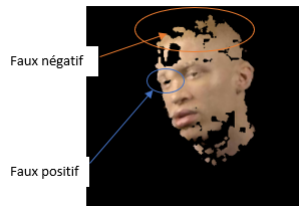


FIGURE 13 – img sortie



FIGURE 14 – img référence

– **(Sur image 18.jpg ./skin-detection 32 0.3 18.jpg perf :70.35%)**

De par ces résultats nous remarquons que le programme détecte bien les pixels peau avec une performance de 70,354 % ce qui est un bon taux, tout de même nous remarquons la présence d'un faux positif (les yeux) qui peut s'expliquer par le fait que le programme ne sait pas ou classer ces pixels vu qu'ils ne sont ni catégorisés dans la base peau ni dans non peau.

– **Expérimentation sur les images hors base d'apprentissage**

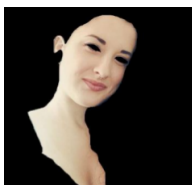
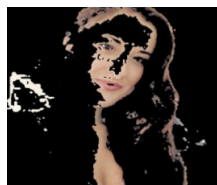
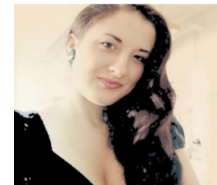


FIGURE 15 – img référence

FIGURE 16 – seuil
0.3 perf = 28.60%FIGURE 17 – seuil
0.2 perf = 37.24%

Au vu de ces résultats nous pouvons noter que les pixels de la peau sont relativement détectés, par contre les mêmes anomalies (détection des yeux, des cheveux, portions des couleurs autres que la couleur de la peau) sont également présente. Aussi nous observons que le seuil de 0.2 appliquer sur la première image de cette section permet d'obtenir une meilleur performance ce qui permet de dire que la variation du seuil influence sur la qualité des images résultats, mais le seuil de 0.3 est mieux adapter pour la majorité de nos images d'apprentissage.

- **Amélioration** Tout au long de notre expérimentation nous avons effectué plusieurs tests sur plusieurs images et avec plusieurs seuils différents, et nous avons obtenu des bon et mauvais résultats. En outre nous avons aussi remarqué la présence des anomalies sur les résultats qui sont due à une mauvaise classification des pixels peau et non-peau dans leurs classes respectifs. Cet anomalie peut être résolue en utilisant une technique d'apprentissage et des règles adéquates pour bien affecter les différents pixels à leurs classes respectives en effectuant des prédictions sur l'appartenance d'un pixel à une classe.

Ainsi pour ce faire une technique de Datamining peut être utilisé pour extraire des règles permettant de prédire si un pixel et peau ou non-peau. Cette technique nous permettra de

construire un modèle de prédiction permettant d'identifier si un pixel est un pixel de peau ou non à partir des différents calculs de probabilité.

4 Conclusion

Au cours de ce TP nous avons implémenté un programme permettant de détecter les régions de peau dans une image. Nous avons créé pour ce faire une base de 30 images différents de par leurs natures, couleurs, luminosités et décors pour créer des images ne faisant que ressortir la peau et d'autres la dissimulant. Ainsi après plusieurs expérimentations réaliser nous retenons que notre programme fonctionne bien car il détecte bien les pixels peau quel qu'en soit leurs couleurs même si nous constatons aussi la présence de certaines anomalies ou des pixels n'ayant aucun trait avec la peau sont considéré peau et des pixels peau considéré comme non-peau. Pour remédier à ces anomalies nous pouvons utiliser des techniques de Datamining combiner avec des techniques de segmentation pour prédire si un pixel est un pixel de peau ou non à partir des calculs de probabilité de peau ou non-peau afin de bien classer les pixels peau et non peau.

La vision par ordinateur est un processus de traitement de l'information Elle utilise des stratégies bien définies afin d'atteindre ses buts L'entrée d'un système de vision est constituée par une séquence d'images Le système lui-même apporte un certain nombre de connaissances qui interviennent à tous les niveaux La sortie est une description de l'entrée en termes d'objets et de relations entre ces objets

5 Référence

1. <https://clouard.users.greyc.fr/Pantheon/experiments/evaluation/index-fr.html>
2. Vezhnevets 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>
3. http://devernay.free.fr/cours/vision/pdf/vision1_intro.pdf
4. <https://openclassrooms.com/fr/courses/1490316-introduction-a-la-vision-par-ordinateur/1490506-premiers-pas>
5. Mohamed HAMMAMI, Youssef CHAHIR, Liming CHEN, Djamel ZIGHED dans l'article Détection des régions de couleur de peau dans l'image.