

Tp1 de vision par ordinateur : détection de la peau

Ce présent rapport a pour objectif la détection de la peau humaine dans une quelconque image. L'idée est d'identifier les pixels qui représentent la peau dans différentes conditions (différentes personnes, différents éclairages). Ainsi, les images que nous utiliserons, doivent être fournies en entrée du programme. Pour cela nous allons passer aux différentes attentes évoqués dans ce TP

1- Détails et fonctionnement du programme

Notre programme implémenté prend en argument l'échelle de réduction des valeurs de l'espace de couleurs, la valeur du seuil de décision et l'image à évaluer. Ce programme fourni en sortie les histogrammes de peau et de non-peau de sa 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.

Pour permettre d'utiliser notre programme, il est important d'effectuer les actions suivantes :

- Composition de notre base d'images :

Dans le répertoire qui accompagne le programme, nous avons stocké un sous-répertoire nommé « base » qui contient à son tour trois sous-répertoires : **peau, non-peau et test**

* **Le premier nommé « peau » : ce sous-répertoire** contient uniquement les images où seuls apparaissent les pixels décrivant la peau (partie non noircie).

* **Le second nommé « non-peau » : ce sous-répertoire** contient uniquement des images où ne figurent pas de pixels peau (partie noircie).

* **Le dernier nommé « test » : ce sous-répertoire** contient les images originales qui ont été traitées pour obtenir les bases peau et non peau. Notons chacun de ces sous répertoires contient 30 images.

- Compilation du programme :

il suffit de se placer dans le répertoire contenant le code source du programme depuis le terminal puis de lancer la commande « make ».

- Exécution du programme :

Il faut lancer la commande suivante: `./detection_peau « echelle » « seuil » « nom_image_test »`. La syntaxe est définie telle que « échelle » correspond à l'échelle de réduction dans l'espace des couleurs, « seuil » est le seuil de décision permettant de classer un pixel comme peau ou non et « nom_image_test » est le nom de l'image fournie en entrée au programme. Exemple : La commande `./detection_peau 32 0.4 1.jpg` permet de détecter les pixels peau dans l'image 1.jpg avec une échelle de 32 et un seuil de décision de 0.4 .

2- Description de l'implémentation

Le processus d'implémentation de notre programme de détection de la peau se décompose en cinq étapes que sont : la création de la base d'images, la construction des histogrammes, la détection de la peau dans les images, le post-traitement et l'évaluation. Ces différentes étapes sont décrites dans les paragraphes ci-dessous.

Création de la base d'images

Notre base d'images a été créée à partir de vingt-cinq (30) images obtenues sur internet. 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 soit aussi différentes que possible afin d'assurer une base d'apprentissage suffisamment diversifiée à notre programme. Une

fois les images sélectionnées, il 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 « peau » et celles l'occultant dans un répertoire nommé « 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 sous répertoires « peau » et « non-peau » qui seront exploitées.

Construction des histogrammes

La construction des histogrammes est une étape très importante dans le processus de détection des pixels peau. Pour notre travail, nous avons besoin de construire deux histogrammes : l'un pour les images « peau » et l'autre pour celles « non-peau ». Dans chaque cas, nous devons d'abord convertir l'ensemble des images de l'espace de couleurs RGB à l'espace de couleur Lab. Ensuite, dans un souci de diminution des calculs à effectuer, nous avons réduit l'espace des valeurs des pixels de $256*256*256$ à $échelle*échelle*échelle$ où « échelle » est la première valeur qui sera mise en paramètre au programme. Par exemple si l'échelle fournie est 32, c'est à dire que, 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 « $256/échelle$ » 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 cases 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 permis la réduction de l'espace des valeurs. Les figures 1 et 2 montrent les histogrammes « peau » et « non-peau » obtenus à partir de notre base d'images.

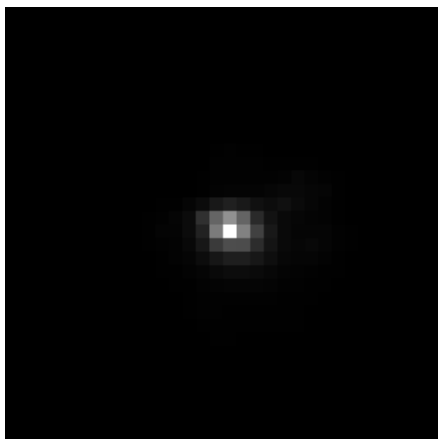


Fig1 : histogramme non peau

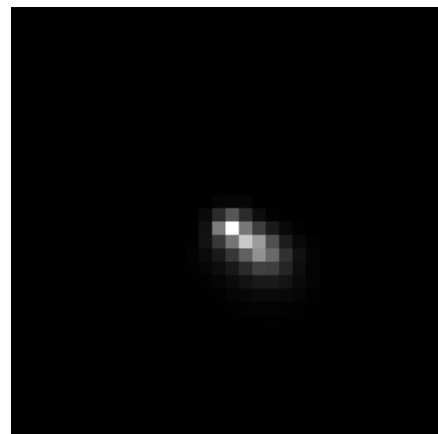


fig2 : peau histogramme peau

Évaluation

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- Expérimentations résultats et analyse

Dans cette partie nous nous consacrerons à la présentation de nos expérimentations, résultats et analyses.

Pour choisir la valeur de seuil à utiliser dans nos expériences nous effectuons une série de tests sur une même image avec différentes valeurs de seuil. Les résultats sont présentés ci-dessous :

Effets du seuil sur la détection de la peau

Les images ci-dessous ont été obtenues en appliquant différents seuils ($0,1 < \text{seuil} < 0,4$) de détection à l'image test 29.jpg.



seuil = 0,1 perf = 20,08, %



seuil = 0,2 perf = 20,08%



seuil = 0,3 perf = 32,18%



seuil = 0,4 perf = 85,59%

On note bien que le meilleur résultat correspond au seuil de valeur 0,4. Nous utiliserons donc la valeur de seuil 0.4 et une échelle de réduction égale à 32 dans la suite de nos travaux.

Détection dans les images de la base d'apprentissage

«./detection_peau 32 0.4 8.jpg » : performance =86,45%



Figure 3 : Image de référence Faux négatifs

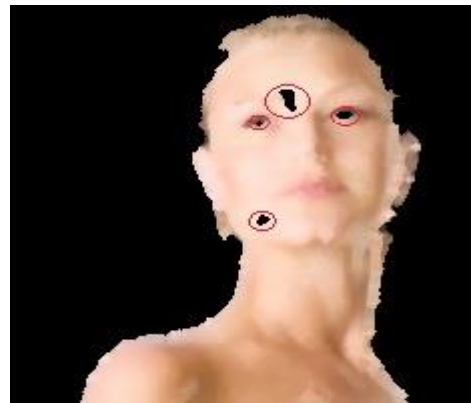


Figure 4 : Image détectée

nous remarquons que : La performance obtenue pour cette image est très bonne car la plupart des pixels « peau » ont été détectés même si l'on note aussi la présence de faux positifs (les yeux de la dame) . On remarque aussi que le programme a bien détecte la peau de couleur blanche. Les yeux sont détectés comme peau car le programme ne sait pas dans quelle catégorie les classer puisqu'ils ne sont ni classé comme peau ou non-peau dans la base d'apprentissage.

. « ./detection_peau 32 0.4 2.jpg » : performance =78,20%



Figure 5 : Image de référence



Figure 6 : Image détectée

Comme nous pouvons constater, pour l'image, le peau de couleur noire est bien détectée . Ajoutons ici que cette peau de couleur noir qui a été détectée signifie que le programme a bien appris les caractéristiques de différents types de peaux.

«./detection_peau 32 0.4 7.jpg » : performance =44,40%



Figure 7 : Image de référence



Figure 8 : Image détectée

Sur l'image de la figure 8 nous remarquons que le programme a bien détecté la peau de couleur blanche qui avait été occultée dans l'image de référence à la figure 7 nous notons la présence des faux positif (cercle rouge) sur quelques parties de la partie de la peau du bébé.

Ce qui justifie la faible performance notée. En outre le programme détecte aussi la table de couleur jaune et des portions du t-shirt de couleur rouge du bébé. Ces détections erronées peuvent se justifier par la faible présence d'éléments de couleur rouge et de de la couleur du bois de la table dans la base d'apprentissage.

«./detection_peau 32 0.4 6.jpg » : performance =78,10 %



Figure 9 : Image de référence



Figure 10 : Image détectée

Comme nous pouvons encore constater, pour la figure10, le peau de couleur blanche a été bien détectée ce qui signifie que le programme a bien appris les caractéristiques de cette peau. Une fois de plus les yeux ont été considérés comme pixels non peau (cercle rouge) avec présence des faux négatifs (cercle bleu).

«./detection_peau 32 0.4 27.jpg » : performance =75,11 %



Figure 11 : Image de référence



Figure 12 : Image détectée

même constat que la précédente à la différence que dans celle-ci, les yeux ont été détectés comme étant peau et qu'il n'y a pas de faux positifs ni négatifs

«./detection_peau 32 0.4 30.jpg » : performance =70,32 %



Figure 13 : Image de référence

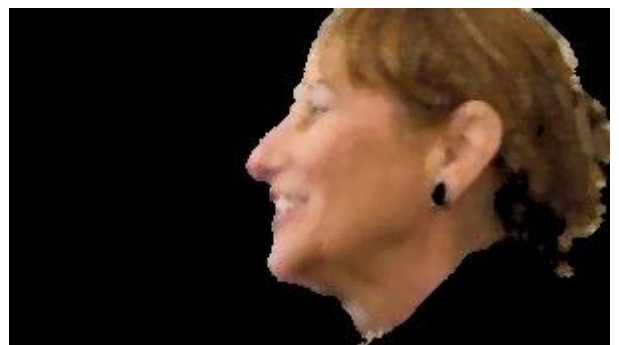


Figure 14 : Image détectée



L'image à la figure14 montre une très bonne détection de la peau et que la boucle d'oreille n'a pas été détecté comme pixels peau. Ainsi, qu'une partie des cheveux ce qui justifie la performance obtenue. Cependant pour les cheveux, nous pensons que c'est le fait que leur couleur tende vers celle de la peau qui justifie leur détection comme pixels peau.

Conclusion de nos expérimentations

A l'issu de ces expériences sur les images de la base d'apprentissage, nous pouvons retenir que la peau est bien détectée même si certaines anomalies sont présentes en raison de quelques insuffisances de caractéristiques dans les images de test(faux positifs et faux négatifs) et de la proximité de certains pixels non-peau avec des pixels peau. Donc si la couleur de peau ou une partie de la couleur de la peau sous l'effet de l'éclairage tends vers le couleur du font de image (background) cette partie ne sera pas détectée et que si la valeur de pixel d'une région ou objet est semblable ou tend vers celle de la peau, ce pixel est possible ou probable d'être détecté comme peau

Conclusion :

De Tout ce qui précède, pour 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