

# Institut de la Francophonie pour l'informatique

## Vision par ordinateur TP 1 : Détection de la peau

**Date de remise : 17/10/2013, 8h30**

**Remise par courriel à : oanhnt@soict.hut.edu.vn**

### Introduction

Durant les travaux pratiques (TPs) en vision par ordinateur, l'objectif sera de développer des algorithmes et des programmes liés à ce qui a été vu en cours. Avec les tps, vous allez comprendre comment construire les outils nécessaires dans ce domaine.

Les tps consisteront en des exercices de programmation en C/C++ en utilisant la librairie **OpenCV** (*obligatoire*) sous **Linux** (*obligatoire aussi*). Vous pouvez choisir toutefois entre la version C ou la version C++ d'OpenCV (*mais pas la version C#, qui ne nous permettra pas d'inclure d'autres librairies*). Les tps se font **individuellement**. Vous pouvez discuter (*en français !*) avec vos voisins, mais chaque personne doit remettre son code et son rapport (*code et rapport différents des autres ! - pas de plagiat*) à la fin. Aucune interface n'est exigée des programmes (*plus facile et plus rapide à programmer*). Vos programmes peuvent être appelés à partir du **shell linux** et prennent les arguments en ligne de commande.

**Lisez attentivement** les consignes de remise des tps qui vous donne plusieurs explications importantes sur le déroulement des tps et l'écriture du rapport. Vous serez évalués suivant les explications et critères donnés sur cette page.

Pour évaluer vos tps, des **demos** peuvent être demandées (en classe) à partir de la date de remise. Conservez bien tous vos tp réalisés (*en état de marche*) pendant tout le semestre pour vérification.

Vous avez un ou plusieurs répertoires d'images disponibles pour chaque TP (voir page du cours). Utilisez seulement ces images dans votre travail (*aucune autre image acceptée*).

### Installation d'OpenCV

Suivez les instructions données à la page :

[http://docs.opencv.org/doc/tutorials/introduction/linux\\_install/linux\\_install.html#linux-installation](http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html#linux-installation)

Sur cette page, vous avez plusieurs liens sur la documentation de la librairie OpenCV. On trouve aussi beaucoup d'autres documentations et exemples sur Internet pour vous aider à apprendre cette librairie facilement.

## Détection de la peau

Dans ce tp, vous allez construire une application de détection de la peau dans les images. Pour cela, il faut tout d'abord construire une base d'exemples de « peau » et de « non-peau » et l'apprendre avec votre programme. Ensuite vous ferez un apprentissage de la couleur de la peau en utilisant l'espace couleurs CIE LAB (composantes A-B seulement).

Pour avoir plus d'informations sur les méthodes de détection de la peau, vous pouvez consulter l'article suivant :

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>

*(voir entre autres la section 2 sur les espaces couleurs et la section 3 sur la construction de modèles de peau)*

Ce tp est directement inspiré de cet article, donc n'hésitez pas à le lire pour y trouver des idées et des explications supplémentaires pour faire ce tp.

Une méthode simple (mais imparfaite) est proposée dans cet énoncé de tp, mais vous avez le droit de prendre une autre méthode plus complète et plus robuste (un bonus de points pourra être donné à ceux qui essaient). Expliquez simplement et clairement vos choix dans le rapport.

Seul le choix de l'espace couleur est imposé dans ce tp : CIE LAB. Le reste est libre, avec comme objectif de bien détecter la peau dans les images.

## Construction d'une base d'images

Pour ce tp, il n'y a pas de base d'images fournie et imposée. Vous allez construire votre propre base d'images, à l'aide de vos propres photos ou tout simplement en cherchant sur Internet. Trouvez entre (minimum) 20 et 30 images en couleurs contenant des personnes (mais pas seulement), afin d'avoir des échantillons contenant de la peau et d'autres choses (non-peau).

**Note** : Pour être correct, il faudrait séparer les images en deux groupes : images d'apprentissage (peau et non-peau) et images de test (validation des résultats). Pour simplifier le travail à faire (identification des pixels à la main), nous utiliserons la même base pour l'apprentissage et les tests.

## Construction d'une base d'exemples peau / non-peau

Il vous faut construire votre base d'exemples, contenant des exemples de peau et des exemples d'autre chose (non-peau). Choisissez des images modèles pour construire votre modèle de la peau. Le choix de bonnes images représentatives est important ici car il influencera la qualité de vos résultats. En utilisant votre éditeur d'images favori (mPaint, gpaint, GIMP ou autres), sélectionnez dans ces images les zones contenant de la peau (visages ou autres). Il suffit de mettre à zéro (à la main en dessinant en noir) tous les pixels de l'image ne représentant pas de la peau (et contraire pour non-peau).

Vous n'avez pas besoin de prendre tous les pixels des images. Prenez des échantillons d'images. Par contre, faites attention à la qualité de vos échantillons. Prenez des exemples de peau dans différentes conditions (différentes personnes, différents éclairages, ...). La quantité et la qualité de vos exemples ici influencera directement la qualité de vos résultats finaux.

Attention : Je vous suggère de trier les pixels en trois classes : peau, non-peau et autres. Vous utiliserez seulement les pixels des deux premières classes et laisserez tomber la troisième. En cas de doute sur une portion d'image qui pourrait perturber vos résultats, classez-la simplement comme « autres ». Ainsi, vous pouvez conserver seulement certaines portions des images « peau » et « non-peau » et pas obligatoirement toutes les images.

## Réduction du nombre de couleurs des images

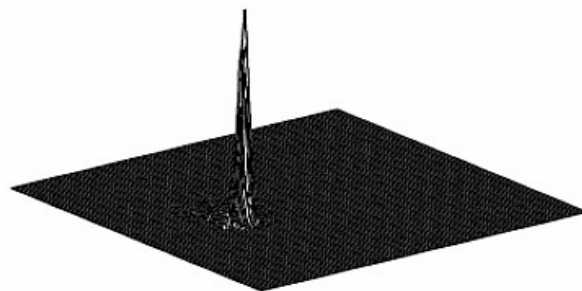
Quelque soit la méthode et l'espace couleur utilisés, il est mieux au préalable de réduire la quantification, c'est-à-dire de réduire le nombre de valeurs pour chaque axe. Par exemple pour RVB, pour chaque dimension, nous réduirons l'échelle des valeurs de 256 à 32 valeurs (32 étant un paramètre que vous pouvez modifier dans vos expérimentations). Une image en RVB 256x256x256 deviendra RVB 32x32x32.

Le même raisonnement est applicable pour tous les espaces couleurs. Dans votre cas, les images sont enregistrés sur disque au format RVB et vous travaillez avec le format LAB. Vous réduirez la quantification en LAB, donc après la conversion depuis RVB.

## Histogramme A-B et modèle de peau

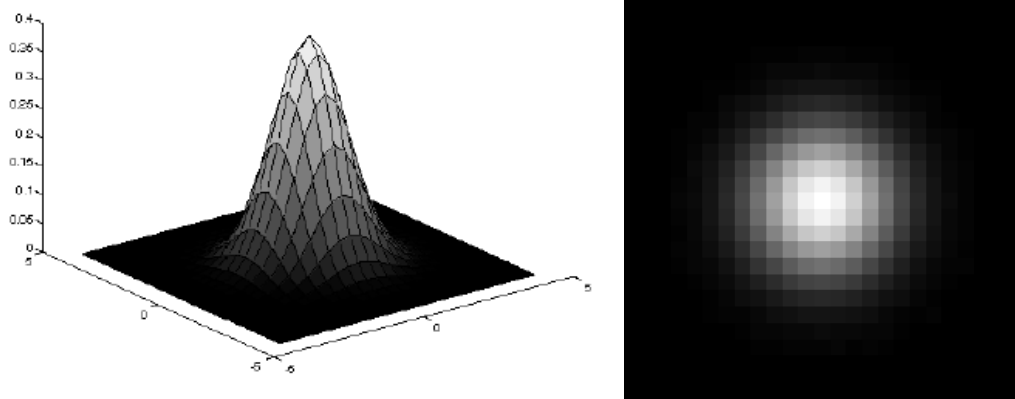
Différents espaces couleurs sont possibles pour détecter les zones de peau dans les images. Dans ce tp, nous allons utiliser l'espace LAB. Vous pouvez utiliser les fonctions fournies dans OpenCV pour convertir une image de l'espace couleur RVB en LAB. Nous n'utiliserons que les composantes A et B (axes de couleurs) et pas L (axe de luminance).

Vous calculerez deux histogrammes 2D de couleur, soit un histogramme « peau » et un histogramme « non-peau ». Un histogramme est un tableau à 2 dimensions (A et B). Pour chaque dimension, nous réduirons l'échelle des valeurs de 256 à 32 valeurs (32 étant un paramètre que vous pouvez modifier dans vos expérimentations). Pour chaque case du tableau, il suffit de compter le nombre de pixels ayant comme valeur un A et un B donnés. Tous les pixels « peau » de toutes les images se retrouvent dans le même histogramme « peau » et réciproquement pour les pixels « non-peau ».



*Histogramme 2D : Nombre de pixels (axe vertical) vs A et B (axes horizontaux).*

Dans votre rapport, vous devez montrer votre histogramme modèle de peau et votre histogramme « non-peau ». Différentes représentations graphiques sont possibles. Choisissez la représentation la plus facile pour vous. Voir exemples de représentation dans la figure suivante.



*Exemple de représentation d'un histogramme. Vous avez ici le même histogramme (même fonction) représenté soit en 3D (à gauche) soit en 2D (à droite). En 2D, le niveau de gris d'un pixel exprime la hauteur z.*

Attention de bien normaliser chaque histogramme. On normalise un histogramme en divisant chaque valeur par le total de pixels. Cela sera utile pour la suite.

Vous obtenez à la fin un « modèle de peau » et un « modèle de non-peau ». Ces modèles génériques sont notés **HistogrammePeau(a,b)** et **HistogrammeNonPeau(a,b)** dans la suite.

Il s'agit d'un modèle de peau assez simple qui peut être amélioré de différentes façons : lissage, dilatation de l'histogramme, etc. N'hésitez pas à développer des idées et à les essayer. Expliquez ce que vous faites dans le rapport. Par exemple, un simple lissage devrait permettre d'améliorer les résultats. Pour une valeur de l'histogramme, vous faites la moyenne des 8 voisins + la valeur elle-même.

## Détection de la peau dans les images

Plusieurs méthodes sont possibles pour détecter la peau dans les images. Lisez l'article cité plus haut pour plus d'informations.

A partir d'un pixel tiré d'une image, une façon simple est de calculer les probabilités de peau et de non-peau à partir de sa couleur  $c$  (a,b).

$$p(\text{peau} \mid c) = p(c \mid \text{peau}) = \text{HistogrammePeau}(a,b)$$

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

La probabilité la plus élevée des deux permet de décider si le pixel est peau ou non-peau.

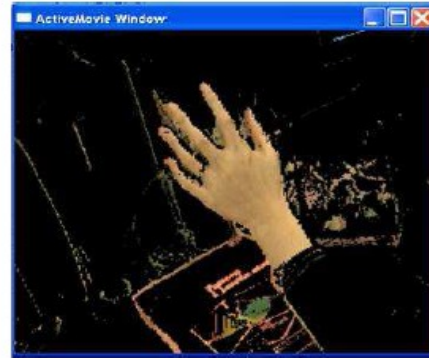
On peut raffiner cette décision en utilisant la méthode de Bayes (équation 12 dans l'article cité plus haut) :

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

où  $p(c \mid \text{peau})$  et  $p(c \mid \neg\text{peau})$  sont données par les histogrammes,  $p(\text{peau})$  et  $p(\neg\text{peau})$  sont les pourcentages de pixels peau et non-peau dans votre base d'apprentissage.

A noter que vous pouvez faire des post-traitements pour améliorer vos résultats (lissage, filtres morphologiques, ...). Ceux-ci ne sont pas exigés pour ce tp. Cependant, si vous utilisez cela, expliquez-le bien dans le rapport cependant.

La figure suivante montre un exemple comment présenter vos résultats dans votre rapport. On montre seulement les pixels détectés peau (à droite) en mettant tout le reste de l'image à zéro (noir). N'oubliez pas de toujours montrer l'image originale avec vos résultats.



*Exemple de résultat de détection de la peau.*

## Rapport à remettre

En utilisant la même base d'images que pour votre apprentissage (simplification de travail, mais non-exacte en terme de résultats obtenus), évaluez le pourcentage de pixels peau que vous réussissez à bien détecter (évaluation quantitative). Montrer dans le rapport des images où la détection fonctionne bien et d'autres où cela fonctionne moins bien, en expliquant pourquoi selon vous.

Ensuite, utilisez quelques (4-5) images autres que celles déjà utilisés, montrez vos résultats sur ces images et expliquez ceux-ci (bons et mauvais aspects) (évaluation qualitative).

Dans l'implémentation de vos algorithmes, vous avez plusieurs choix à chacune des étapes, que ce soit pour les images utilisées ou la programmation de l'algorithme. Tout n'est pas précisé dans cet énoncé. Veillez à bien présenter chaque étape de votre démarche dans le rapport et à bien expliquer chacun de vos choix. Montrez des exemples à chaque étape.

Montrez vos images modèles pour la phase de construction du modèle de peau, votre sélection de régions de peau, ainsi que vos histogrammes modèle **HistogrammePeau(a,b)** et **HistogrammeNonPeau(a,b)** sous forme d'images (ou de graphiques).

Comment peut-on améliorer les résultats que vous avez obtenus ? Donnez des idées.

N'hésitez pas à suggérer et à tester de nouvelles idées ici. Plusieurs variantes améliorant les résultats sont possibles.

Bon travail !