

TD 4

« utilisation d'une table de LUT – stretching automatique »

1. Utilisation d'une table de LUT

a. Expression d'affectation à travers une LUT

Par quelle instruction en langage C peut-on affecter au pixel (ichannel,ili,ipx) de l'image traitée stockée dans le tableau processed_image, la valeur d'une fonction globale de transformation radiométrique appliquée au même pixel (ichannel,ili,ipx) de l'image originale et stockée dans le tableau origin_image.

C'est pour cette raison que les transformations radiométriques globales sont parfois appelées « fonctions de LUT » !

b. Fonction de transformation radiométrique globale

Pourquoi ces fonctions $F(r)$ appliquées aux valeurs radiométriques r sont-elles globales ?

c. Fonction de transformation d'histogramme

Soit $H(r)$ les valeurs de l'histogramme d'une image,
soit $F(r)$ la fonction radiométrique appliquée à cette image,

écrire le programme dans un pseudo-code qui calcule l'histogramme $H_F(r)$ de l'image après traitement par la fonction F .
Ce calcul sera réalisé en se servant uniquement de $H(r)$ et de $F(r)$ et donc sans parcourir l'image de sortie.

C'est pour cette raison que les transformations radiométriques globales sont plus communément appelées « fonctions de transformation d'histogramme » !

d. Menu des traitements radiométriques globaux

Ecrire un programme implémentant différentes transformations radiométriques globales (1-Identique, 2-Opposée, 3-Seuillage à 128, 4-Stretching linéaire défini par les 4 points (0,0) (50,0) (205,255) (255,255), 5-Stretching linéaire défini par les 2 points (0,50) (255,205), 6-...) que l'utilisateur pourra choisir en fournissant un paramètre supplémentaire sur la ligne de commande.

Ce programme devra faire l'économie de boucles répétées.

Quelle est la différence entre la transformation 4 et la 5 ?

2. Stretching automatique

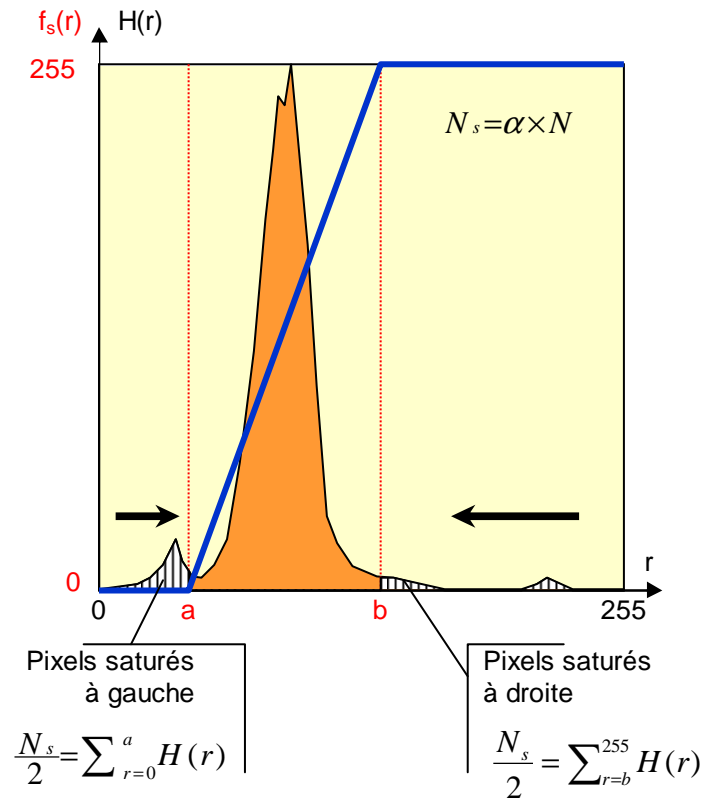
On cherchera ici à écrire un programme permettant à l'utilisateur de fournir le « pourcentage de saturation » α et qui calculera automatiquement les bornes a et b du stretching.

a. Recherche à partir d'un histogramme

Soit $H(r)$ les valeurs de l'histogramme d'une image.

Ecrire le programme dans un pseudo-code qui calcule le nombre total de pixels image (N) à partir des valeurs de l'histogramme $H(r)$.

Ecrire le programme dans un pseudo-code qui calcule la valeur de la borne a en fonction des valeurs de l'histogramme $H(r)$, de α et de N .



Ecrire le programme dans un pseudo-code qui calcule la valeur de la borne b en fonction des valeurs de l'histogramme $H(r)$, de α et de N .

b. Application aux images

Ecrire le programme de « stretching automatique » et l'appliquer aux trois canaux des images de roissy et san-remo. Rappeler ci-dessous les valeurs des bornes de stretching trouvées automatiquement par le programme pour les différentes valeurs de « pourcentage de saturation » α .

roissy				san-remo			
$\alpha = 0,5 \%$		$\alpha = 1,5 \%$		$\alpha = 0,5 \%$		$\alpha = 1,5 \%$	
$a_1 = \dots\dots\dots$	$b_1 = \dots\dots\dots$	$a_1 = \dots\dots\dots$	$b_1 = \dots\dots\dots$	$a_1 = \dots\dots\dots$	$b_1 = \dots\dots\dots$	$a_1 = \dots\dots\dots$	$b_1 = \dots\dots\dots$
$a_2 = \dots\dots\dots$	$b_2 = \dots\dots\dots$	$a_2 = \dots\dots\dots$	$b_2 = \dots\dots\dots$	$a_2 = \dots\dots\dots$	$b_2 = \dots\dots\dots$	$a_2 = \dots\dots\dots$	$b_2 = \dots\dots\dots$
$a_3 = \dots\dots\dots$	$b_3 = \dots\dots\dots$	$a_3 = \dots\dots\dots$	$b_3 = \dots\dots\dots$	$a_3 = \dots\dots\dots$	$b_3 = \dots\dots\dots$	$a_3 = \dots\dots\dots$	$b_3 = \dots\dots\dots$

Les images et les résultats seront présentés dans un rapport HTML.