

Compte-rendu TP06 MI01

Le but de ce TP et des suivants sera donc de développer petit à petit en assembleur un logiciel de détection de contours dans une image, en utilisant un algorithme simple.

Conversion en niveaux de gris

La conversion en niveaux de gris / d'un pixel RVB se fait selon la formule suivante :

$$I = R \times 0,299 + V \times 0,587 + B \times 0,114$$

Nous devons donc convertir les coefficients décimaux en hexadécimal et les avons décalés de 8 BITS vers la gauche.

Bleu:

$$0,114 * 2 = 0,228 : 0$$

$$0,228 * 2 = 0,456 : 0$$

$$0,456 * 2 = 0,912 : 0$$

$$0,912 * 2 = 1,824 : 1$$

$$0,824 * 2 = 1,648 : 1$$

$$0,648 * 2 = 1,296 : 1$$

$$0,296 * 2 = 0,592 : 0$$

$$0,592 * 2 = 1,184 : 1$$

Vert:

$$0,587 * 2 = 1,174 : 1$$

$$0,174 * 2 = 0,348 : 0$$

$$0,348 * 2 = 0,696 : 0$$

$$0,696 * 2 = 1,392 : 1$$

$$0,392 * 2 = 0,784 : 0$$

$$0,784 * 2 = 1,568 : 1$$

$$0,568 * 2 = 1,136 : 1$$

$$0,136 * 2 = 0,272 : 0$$

Rouge:

$$0,299 * 2 = 0,598 : 0$$

$$0,598 * 2 = 1,196 : 1$$

$$0,196 * 2 = 0,392 : 0$$

$$0,392 * 2 = 0,784 : 0$$

$$0,784 * 2 = 1,568 : 1$$

$$0,568 * 2 = 1,136 : 1$$

$$0,136 * 2 = 0,272 : 0$$

$$0,272 * 2 = 0,544 : 0$$

Ce qui donne en hexa :

0001 1101

01 Dh

1001 0110

09 6h

0100 1100

04 Dh

Traitement d'image – Première partie

La boucle consiste à manipuler les octets du rouge, vert, et bleu d'un séparément au moyen de trois registres différents et à les additionner dans un registre unique. On sauvegarde en fin de boucle. Chaque itération traite un pixel de l'image.

2

Compte-rendu TP06 MI01

| | |
|--|--|
| <pre>ADD EAX, EBX SHR EAX, 8 MOV [EDI + ECX*4 - 4], EAX DEC ECX CMP ECX, 0 JNE suivant ;***** fin: pop edi pop esi pop ebx pop ebp ret ; Retour à la fonction MainWndProc process_image_asm ENDP END</pre> | <p>Addition</p> <p>Sauvegarde du nouveau pixel Décrémentaton du compteur Si compteur égale à 0 on jump</p> |
|--|--|

Compte-rendu TP06 MI01

Conclusion:

Ce TP nous a une fois de plus amené à manipuler les sauts conditionnels. Nous avons également été amenés à prêter attention à la taille des registres que nous utilisons en fonction de des mnémoniques utilisées. Enfin nous avons dû utiliser à bon escient les décalages à droite afin de pouvoir réaliser les additions comme il faut et les multiplications sans déborder. Evidemment nous avons dû faire attention à employer aussi peu d'accès mémoire que possible.