

## **I) Introduction :**

Dans ce document, nous avons synthétisé le travail effectué au cours de notre projet. Le but de ce projet était d'implémenter une fonctionnalité simple de traitement d'images : la détection de contour d'une image. Nous devons utiliser pour cela le filtre de Sobel. Durant cette synthèse nous avons expliqué en détails le déroulement du projet, notamment quelles méthodes nous avons employé afin de le réaliser, les problèmes auquel nous avons fait face, et comment nous nous sommes réparti les tâches.

## **II) Déroulement du projet :**

### **1.1 Analyse et compréhension du sujet**

Afin de débiter le projet, la première chose à faire a été de lire et d'analyser en détails notre sujet. Comme tout n'était pas compris, nous avons décidé d'effectuer d'autres recherches afin d'approfondir nos connaissances sur le filtre de Sobel, les commandes de l'assembleur, et certains mots incompris dans le sujet comme ce qu'était le seuillage. Notre première difficulté a donc été de comprendre le sujet et de savoir comment débiter notre projet. Pour nous aider, nous avons également répondu aux questions dans le sujet afin de nous éclairer un peu et avons par la suite commencé la programmation.

### **1.2 Codage et explications des choix**

Nous avons créé plusieurs versions de notre code afin de pouvoir revenir en arrière en cas de problème et décortiqué partie par partie ce qu'il fallait faire et comment il fallait le faire.

#### **(Saisie du fichier)**

Tout d'abord, nous avons commencé par coder la partie où on demandait à l'utilisateur d'entrer une chaîne de caractère (un fichier) au format bitmap(.bmp). Nous avons mis cette chaîne de caractère dans une variable puis nous avons créé une fonction qui enlevait le slash n à la fin du nom du fichier pour qu'il puisse être reconnu sans erreur. Nous avons préféré utiliser des noms de variables au lieu des registres de stockages (s0, s1...) car c'était plus pratique et plus lisible au niveau du code.

#### **(Ouverture du fichier saisi)**

Dans un second temps, nous avons créé une fonction d'ouverture du fichier entré par l'utilisateur avec gestion d'erreur. La fonction de gestion d'erreur déterminait si la chaîne de caractère était un fichier ou non. Si ce n'était pas un fichier, alors le programme affichait une erreur. Si c'était un fichier on devait l'ouvrir et créer une variable dans laquelle on mettait le nom du fichier en modifiant le nom et en rajoutant Contour dans le nom de base

**(Lecture de l'entête du fichier)** Nous avons ensuite alloué 2 octets afin de récupérer les 2 octets du format bmp (l'identification 42 4d qui permet de dire que le format du fichier est au format bitmap) puis nous avons créé une fonction afin de pouvoir vérifier si notre fichier était bien au format bmp ou non. Si le fichier était au bon format alors le programme devait continuer sinon une erreur était renvoyée. Nous avons ensuite alloué 4 octets pour la lecture de la taille totale du fichier que nous avons stocké dans une variable puis réalisé un décalage de 4 octets et enfin alloué encore 4 octets afin d'effectuer la lecture de l'offset de l'image.

**(Lecture de l'entête de l'image)** Nous avons effectué un décalage de 4 octets puis réalisé la lecture de la largeur et la hauteur de l'image en pixels, un nouveau décalage de 2 octets et enfin réalisé la lecture du nombre de bits utilisés pour coder la couleur et stockés ces valeurs dans des variables.

#### **(Traitement de l'image)**

Après avoir stocké ces informations de l'image principale, nous avons refermé notre fichier bitmap puis ouvert 2 autres fichiers. L'un pour l'entrée et l'autre pour la sortie de l'image.

Nous sommes ensuite passés à l'écriture de la fonction écriture de l'en-tête de l'image dans notre image de sortie dans laquelle nous avons alloué la mémoire puis effectué une lecture de l'offset et enfin réalisé l'écriture de cet offset dans l'image de sortie.

Nous avons réalisé une allocation de la mémoire pour la taille totale de l'image, la matrice Fx et la matrice Fy, et avons soustrait l'offset de la taille totale de l'image et l'avons stocké. Enfin, nous avons copié l'image originale dans le registre s5 et avons refermé notre fichier bitmap.

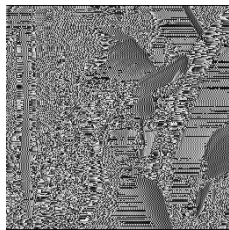
#### **(Ecriture de l'image)**

Nous avons créé deux fonctions, pour déterminer le nombre de pixel en largeur et en hauteur de l'image afin de réaliser le traitement. Nous avons ensuite créé deux tableaux, un pour la matrice Gx et l'autre pour la matrice Gy. Puis effectué un décalage de la ligne 255 pour le tableau Gx. Dans notre programme, nous

n'avons finalement pas utilisé le tableau de Gy. Nous avons réalisé une fonction pour le filtre de Sobel qui parcourt la hauteur et la largeur de l'image et qui utilise ensuite les matrices Fx et Fy afin de trouver le résultat de Gx et Gy dans une même boucle. De plus, nous avons fait en sorte d'additionner GX et GY directement dans la fonction ce qui nous a permis de réaliser le filtre en une fonction. Nous avons également bien pris en compte le seuillage et la valeur absolue des résultats négatifs.

### **1.3 Difficultés rencontrées**

La réalisation de ce projet a été complexe. Une des difficultés rencontrées durant ce projet a été de ne pas vraiment savoir si ce que nos fonctions renvoyaient était ce qui était attendu. Nous avons donc dû réaliser des fonctions tests qui nous permettaient de savoir si nos programmes renvoyaient les bons résultats ou non. Comme vu dans la partie [1.1 Analyse et compréhension du sujet](#), la première difficulté que nous avons rencontrée a été de comprendre le sujet. Nous avons mis un peu de temps avant de savoir comment commencer le code. Après de nombreuses heures passées à réfléchir nous nous sommes mis à coder. La partie traitement de l'image a été la plus complexe à coder. Nous avons commencé par refaire une analyse du sujet et déterminer quelles fonctions nous devons faire et ce que nous devons utiliser dans les fonctions. Nous savions qu'il fallait utiliser la matrice A de taille N x M de notre image de départ ainsi que la matrice F de taille 3x3 (plus précisément les matrices Fx et Fy) afin de réaliser le produit de convolution de A et F (soit A et Fx, et A et Fy dont les résultats seraient Gx et Gy). Puis qu'il fallait réaliser une fonction valeur absolue afin de faire la valeur absolue de Gx et Gy. Puis réaliser l'addition de ces deux valeurs absolues et réaliser ensuite un seuillage de cela. Au fur-et-à mesure des informations complémentaires se rajoutaient pour rendre notre résultat plus conforme au résultat attendu et pour nous donner un résultat tout court. Nous avons peiné à trouver comment mettre tout cela sous forme de code. Petit à petit et en posant des questions à nos professeurs nous nous sommes débloqués et avons pu avancer. Le résultat de notre image avec Contour s'est formé étape par étape.



Premier résultat



Second résultat



Troisième résultat

Et enfin nous avons réussi à former le résultat final de l'image notre code :



### **1.4 Répartition des tâches**

Nous avons commencé par analyser le sujet ensemble. Nous avons ensuite fait des recherches individuelles pour approfondir notre compréhension du sujet et une fois le sujet compris, nous avons réfléchi à plusieurs solutions de code chacun de notre côté puis nous avons réalisé le code ensemble en utilisant et mixant ces différentes solutions trouvées. Enfin, nous avons effectué la synthèse du projet.

## **III) Conclusion :**

En conclusion, ce projet nous a demandé beaucoup d'heures de travail et de réflexion. Il n'a pas été simple car la compréhension du sujet était assez complexe et que le codage des fonctions était dur mais avec de la patience, beaucoup de temps et travail ainsi que de la motivation nous sommes parvenues à terminer notre projet. Il fut très intéressant car le sujet était attrayant et qu'il nous a permis d'évoluer dans le domaine de l'assembleur MIPS, tout en travaillant nos capacités d'analyse, de recherches, et de compréhension des consignes.