Traitement image :  
  
Introduction :

La partie traitement image se concentre sur l'analyse d'une image représentée sous forme de matrices RGB d’entier. Le programme est implémenté en langage C et comporte plusieurs fonctions pour effectuer des opérations telles que la recherche et le stockage des plages de couleur (bleu ou rouge), le comptage des occurrences dans ces plages, et la détermination des valeurs minimales et maximales des lignes et colonnes.

Une image contenant texte, capture d’écran, Rectangle, carré

Description générée automatiquement

Fichier texte fourni comportant chaque valeur RGB en entier

Structures de Données :

Le programme utilise deux structures principales, Range pour stocker les informations sur les plages de couleur bleue, et InformationsImage pour stocker les données générales sur l'image ainsi que les résultats des opérations. Ces structures sont cruciales pour organiser et manipuler les données de manière efficace.

Fonctions Principales :

masquerDeuxBitsPoidsFortEnBinaire:

Cette fonction prend une valeur entière en entrée et renvoie une chaîne binaire représentant les deux bits de poids fort de cette valeur et le fait pour chaque composantes (rouge vert et bleu).

Une image contenant capture d’écran, ligne, Rectangle, texte

Description générée automatiquement

Matrice quantifiée

On quantifie sur 2 bits chaque composantes RGB donc on arrive avec une valeur sur 6 bits (ex : 110100) les deux bits de poids forts (11) représentent la composante rouge, les deux bits du milieu (01) représentent la composante verte et les deux bits de poids faibles (00) représentent la composante bleue.

enregistrerPlagesB :

Crée et écrit les informations sur les plages de couleur bleue dans un fichier texte.

trouverEtStockerPlageB :

Identifie et stocke les plages de couleur bleue dans la structure InformationsImage.

compterDansPlageB :

Compte le nombre d'occurrences de la couleur bleue et de 'N’ dans chaque plage de couleur bleue.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

trouverMinMaxPlagesB :

Trouve la plus petite et la plus grande valeur de ligne et de colonne avec la couleur bleue, calculant également la moyenne des occurrences de la couleur bleue dans l'image. Si la forme est trop petite elle n’est pas reconnue comme une balle

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

traiterImage :

Utilise les fonctions précédentes et effectue un traitement sur la matrice quantifiée afin de la transformée en une matrice de caractère

Une image contenant capture d’écran, motif, Rectangle, ligne

Description générée automatiquement

Dans cette image les pixels en surbrillance sont les caractères ‘B’ et cette fonction renvoie s’il y a ou non une balle sur l’image et sa couleur (bleu ou rouge)



**Conclusion sur les Limitations du Code :**

Bien que le code implémenté soit capable de détecter et d'analyser les plages de couleur bleue dans une image, plusieurs limitations significatives sont à noter.

Reconnaissance de Formes :

Une des limitations majeures est l'absence de reconnaissance de formes. Le programme se contente de détecter des plages de couleur bleue, mais il ne peut pas distinguer si ces plages forment réellement une forme spécifique telle qu'une balle. Ainsi, il peut donner des résultats erronés en identifiant des zones de couleur bleue qui ne correspondent pas nécessairement à une balle.

Exemple :

Une image contenant capture d’écran, Rectangle, gris, sol

Description générée automatiquement

Limitation aux Couleurs Bleue et Rouge :

Le code est spécifiquement conçu pour traiter les plages de couleur bleue, avec une détection basée sur des seuils définis. Bien que cela soit adapté à des cas spécifiques, cela exclut la possibilité de traiter d'autres couleurs qui pourraient être pertinentes dans divers scénarios.

Quantification sur 2 Bits :

La quantification sur seulement 2 bits pour représenter les couleurs dans les plages de couleur bleue peut conduire à une perte de distinction entre certaines nuances. Notamment, la couleur rouge et la couleur jaune sont traitées de manière équivalente dans ce schéma, ce qui peut entraîner des confusions dans l'analyse des couleurs.

Stockage de l'Information sur l'Obstacle :

Conformément aux spécifications du dossier, le code ne stocke pas la position précise de l'obstacle détecté. Il se limite à identifier la plage de couleur bleue sans fournir d'informations détaillées sur la position ou la forme de l'obstacle, ce qui peut être crucial pour une application réelle.

**Perspectives d'Amélioration :**

Pour améliorer ces limitations, des modifications et des ajouts peuvent être envisagés :

Intégration de la Reconnaissance de Formes :

L'ajout d'algorithmes de traitement d'image plus avancés pour la reconnaissance de formes permettrait au programme de distinguer les objets en fonction de leur forme, plutôt que de se fier uniquement à la couleur.

Augmentation de la Précision de la Quantification :

Une augmentation du nombre de bits utilisés pour la quantification des couleurs permettrait une représentation plus précise des nuances de couleurs, évitant ainsi les confusions entre le rouge et le jaune.

Stockage Précis des Informations sur l'Obstacle :

Pour une application pratique, il serait nécessaire de stocker des informations plus détaillées sur l'obstacle, telles que sa position précise dans l'image, sa forme, et éventuellement d'autres caractéristiques pertinentes.

En Conclusion :

Le code actuel constitue une première étape dans le traitement des images en identifiant les plages de couleur bleue, mais il nécessite des améliorations pour une application plus sophistiquée de reconnaissance d'objets. Les suggestions d'améliorations mentionnées ci-dessus visent à renforcer la capacité du programme à analyser et interpréter de manière plus précise les caractéristiques des objets dans une image.