

Correction automatique de la couleur

1st Cristian Chávez
chavez@telecom-paris.fr

2nd Juan Rios
juan.riosgallego@telecom-paris.fr

I. DESCRIPTION DU SUJET

L'objectif de ce projet c'est d'implémenter l'algorithme "Automatic Color Equalization" présente dans les articles "A new algorithm for unsupervised global and local color correction", Rizzi, A., Gatta, C., Marini, D., Pattern Recognition Letters, 24(11), pp. 1663-1677, 2003 et "Automatic Color Enhancement (ACE) and its Fast Implementation", P. Getreuer, IPOL, 2012. Le but de l'algorithme c'est modéliser parmi les méthodes de "Gray World" et "White Patch" le système de vision humaine pour adapter des images avec conditions d'éclairage très variables et d'extraire l'information de l'environnement.

L'algorithme cherche conserver la constance des couleurs et le constance de la luminosité

Le premier pas dans le projet c'est réaliser l'ajustement chromatique à travers de la comparaison chromatique, l'équilibrage local / global et définir la sélection des pixels

Dans une deuxième part, il faut faire l'échelle de la reproduction dynamique des tons, à travers des autres algorithmes comme "Gray World" ou le but est que la moyenne des valeurs des canaux de couleur soit un ton de gris neutre (généralement gris moyen), l'algorithme de la méthode "Gray World" calcule la valeur de la moyenne pour chaque canal de couleur et ajuste ensuite les valeurs des canaux de couleur pour que la moyenne soit la même pour chaque canal et aussi le méthode "White Patch", il s'agit d'une technique de correction de l'équilibre des blancs qui permet de neutraliser les couleurs d'une image de la manière dont l'objet le plus brillant dans la scène apparaît complètement blanc.

II. TRAVAUX DÉJÀ RÉALISÉS

- L'implémentation de l'algorithme original propose par les auteurs Rizzi, A., Gatta, C., Marini, D. en utilisant l'environnement Google - Colab et le langage python.
- Méthode pour l'ajustement chromatique et spatial : Nous avons utilisé des photos qui n'avaient des valeurs qu'en niveaux de gris et des images. Pour la fonction de distance $d(\cdot)$, nous avons choisi la distance euclidienne, qui donne de bons résultats, comme le montre l'histogramme, car elle permet d'obtenir une répartition des couleurs plus uniforme. Nous avons testé différentes tailles pour le sous-ensemble utilisé dans la première partie de l'ajustement chromatique/spatial, car si nous utilisons l'image entière comme sous-ensemble, comme l'ont fait les auteurs dans leur travail original, l'algorithme devenait de complexité $O(N^4)$, ce qui entraînait

des retards significatifs lors des tests de l'algorithme au passage de 100x100 à 200x200 pixels (ce qui représente toujours une résolution très basse). Lorsque la taille du sous-ensemble utilisée était constante et inférieure à celle de l'image, nous pouvions obtenir des résultats plus rapides. Pour une taille de sous-ensemble nettement inférieure à celle de l'image, le problème devenait de complexité $O(N^2)$, ce qui nous permettait de réaliser des tests plus rapidement, bien qu'avec une moindre qualité, car l'ajustement chromatique/spatial ne tenait pas compte de tout l'espace ni de la couleur de l'image, mais le faisait de manière locale pour chaque pixel. Nous avons également choisi la fonction de saturation avec une pente de 20 pour la fonction $r(\cdot)$, car elle nous a donné les meilleurs résultats, car la fonction signum générait trop de contraste dans l'image résultante R.

- Méthode pour la mise à l'échelle de la reproduction dynamique des tons : Après avoir terminé la première partie, qui était la plus complexe en termes informatiques, nous passons à une deuxième phase. Au départ, nous avons mis en œuvre la mise à l'échelle linéaire, ramenant ainsi R dans la plage $[0, 255]$. Cependant, la mise à l'échelle par White Patch ou Gray World, qui ramenait également R dans la plage d'une image, produisait de meilleurs résultats, conformément à ce que dit l'article, en centrant l'image finale sur un gris moyen.
- Les test de validation (tables statistiques, histogrammes de comparaison et la constance des couleurs)
- L'implémentation de l'algorithme pour les images qui ont 3 canaux des couleurs
- L'implémentation de l'algorithme pour l'amélioration local, avec seulement la sélection d'un petit groupe des pixels contigus pour chaque iteration

III. TRAVAIL À ACCOMPLIR

- Implémentation de l'algorithme avec l'amélioration de vitesse
- La méthode d'approximation polynomiale pour la fonction de pente sera utilisée pour décomposer l'opération principale en convolutions et la méthode d'interpolation sera utilisée, les deux méthodes réduisent la complexité du problème à $O(N^2 \log N)$