Rapport TP2 - Analyse d'images

Yanisse FERHAOUI - Amine ITJI

Lien du dépot :
https://forge.univ-lyon1.fr/p1909519/mif17-analyse-d_images/-/tree/main/TP_2

I. Transformation de Hough

La transformation de Hough est une technique fondamentale en traitement d'image pour la détection de formes géométriques telles que les lignes et les cercles. Son principe repose sur la représentation des formes à détecter dans un espace paramétrique, où chaque paramètre de la forme correspond à une dimension de cet espace.

Dans le cas de la détection de lignes, chaque ligne est représentée par deux paramètres : ρ (rho) et θ (theta). ρ représente la distance de la ligne par rapport à l'origine (0,0), mesurée perpendiculairement à la ligne, tandis que θ représente l'angle formé par la ligne et l'axe horizontal, généralement mesuré en radians dans le domaine [-π/2, π/2].

Pour détecter des lignes dans une image, on utilise un accumulateur de Hough. Pour chaque pixel de l'image, toutes les valeurs possibles de ρ et θ pour lesquelles ce pixel appartient à une ligne sont calculées. Ces valeurs de ρ et θ forment un espace appelé l'espace Hough, où chaque pixel est un accumulateur. Les accumulateurs correspondants dans l'espace Hough sont incrémentés pour chaque pixel de l'image.

Une fois que tous les pixels de l'image ont été analysés, on recherche dans l'espace Hough les valeurs de ρ et θ qui ont accumulé le plus de votes. Ces valeurs correspondent aux paramètres de la ligne détectée.

Pour la détection de cercles, chaque cercle est défini par trois paramètres : le centre (x, y) et le rayon (r). L'espace Hough dans ce cas est tridimensionnel, avec des axes correspondant aux coordonnées (x, y) du centre et au rayon r du cercle. Le processus est similaire à celui de la détection de lignes, mais au lieu d'accumuler des votes pour des lignes, on accumule des votes pour des cercles.

En résumé, la transformation de Hough convertit le problème de la détection de formes dans l'image en un problème de détection de points dans un espace paramétrique. Cette représentation permet une détection robuste des formes géométriques, indépendamment de leur orientation, taille ou position dans l'image.

II. Détections de lignes

La détection de lignes est une étape importante de notre TP. Nous avons ré-implémenté la fonction ***HoughLinesP***, qui prend en paramètres une image binaire, le vecteur de couples de points que cette fonction remplira, ainsi que les détails tels que le seuil, le rhô et theta, et la taille minimum et maximum des segments à conserver. Cette fonction parcourt toute l'image, pour chaque pixel de valeur supérieure à 0 on remplit l'accumulateur pour toutes les données polaires correspondantes à ce pixel, puis un vote pondéré est fait pour déterminer les coordonnées polaires à conserver. Ces données sont

ensuite transformées en données cartésiennes, puis insérées dans le vecteur de couples de points en fonction de la taille minimum et maximum des segments. Nous utilisons ensuite les vecteurs de couples de points pour tracer les droites correspondantes dans la fonction **drawLines**. Avec chaque droite de couleur différente.

Nous constatons que selon les images, le seuil varie énormément, il faut être le plus précis possible et changer régulièrement le seuil pour obtenir un résultat optimal.

III. Détections de cercles

La détection de cercles est cruciale pour notre projet. Nous avons ré-implémenté la fonction ***`HoughCircles`*** de opencv, celle-ci prend en paramètre notre image binaire, un tableau de vecteur 3D qui correspondra aux cercles de coordonnées (a, b) et de rayon r qu'il faudra tracer. Dans cette fonction nous parcourons l'image, pour chaque pixel, nous parcourons chaque zone qui englobe le rayon sélectionné, ensuite nous incrémentons l'accumulateur à la case (a, b, r) lorsque l'équation $(x-a)^2+(y-b)^2=r^2$. Une fois cela fait, un vote pondéré est réalisé en fonction du seuil et les cercles à tracer sont envoyés dans le tableau de vecteur 3D. Enfin, les cercles sont dessinés grâce à la fonction **drawCircles**.

Nous remarquons que cette fonction, selon l'intervalle du rayon minimum et maximum, et en fonction du nombre de contours dans l'image, peut devenir très coûteuse.

IV. Résultats

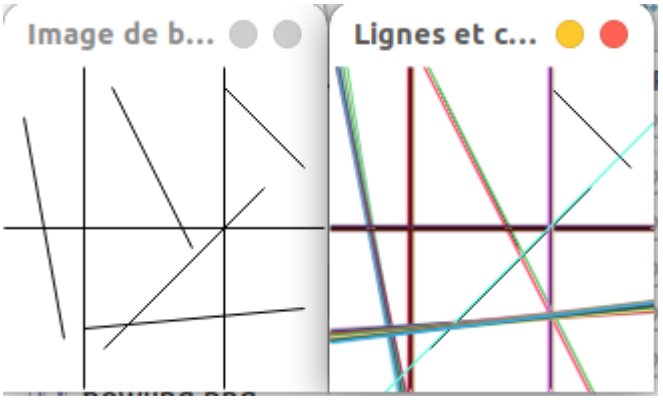

Figure 1 : Capture d'écran détection de lignes de l'image Droites_simples.png avec un seuil à 100

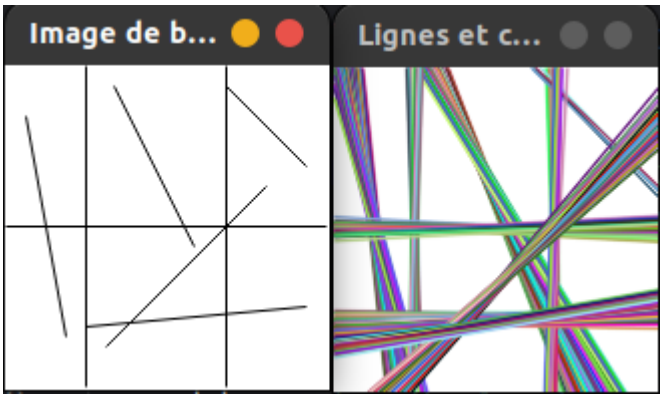

Figure 2 : Capture d'écran détection de lignes de l'image Droites_simples.png avec un seuil à 50

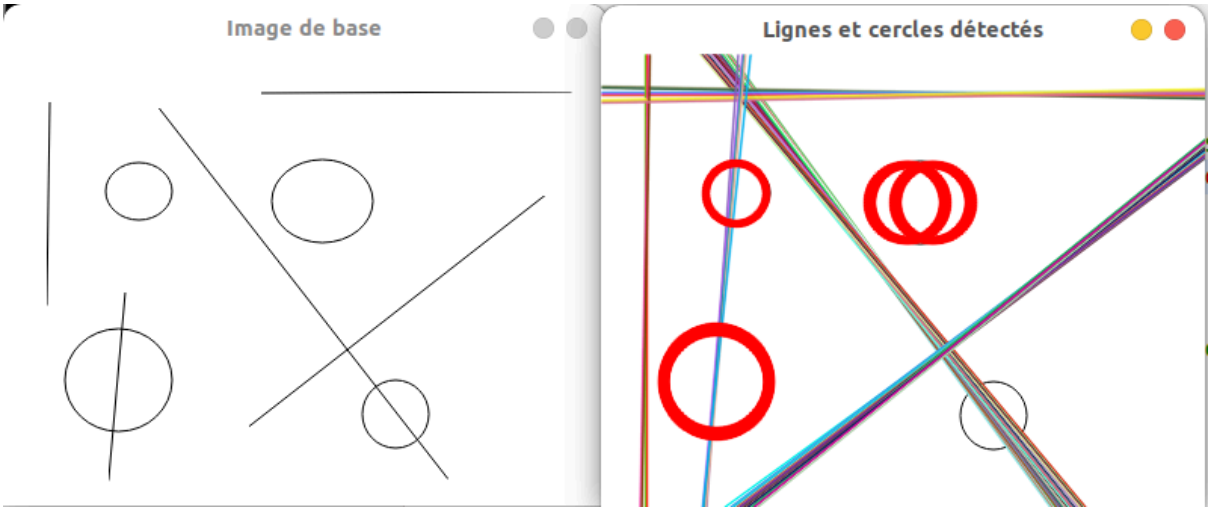

Figure 3 : Capture d'écran détection de cercles de l'image droites_et_cercles.png

Les 2 cercles sont "collés" dans l'image du dessus car c'est un cercle de forme ovale qui a été détecté.

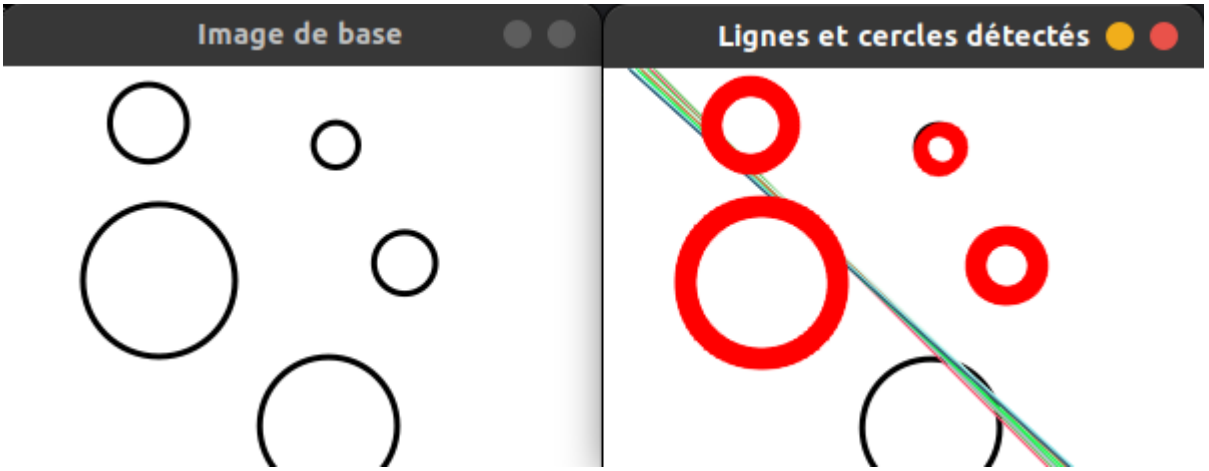

Figure 4 : Capture d'écran détection de cercles de l'image cercles.png

À noter que des lignes sont tracées car des points appartenant à différents cercles sont alignés et que le seuil utilisé dans cette détection de lignes est faible.