UQAC

TP 1 - Vision artificielle et traitement des images

Algorithme de detection

Benoît Faure FAUB20060100
Basil Courbayre COUB30119800
Ali Ghammaz GHAA27070100
Octobre 2023

Table des matières

1 Introduction

L'objectif de ce travail pratique est de détecter les changements dans un environnement composé de trois pièces distinctes d'un appartement : la cuisine, le salon et la chambre. Pour chaque pièce, nous disposons d'une image de référence, représentant l'état initial de la pièce, et d'images ultérieures qui présentent des changements dans l'environnement, principalement au niveau du sol. Le défi consiste à concevoir un algorithme capable de détecter ces changements dans les trois pièces en traitant des images prises dans des conditions d'éclairage variées.

Ce rapport détaille les étapes du traitement d'image mises en œuvre pour relever ces défis, les méthodes et algorithmes employés, ainsi que les résultats obtenus. Il met également en évidence l'importance de la détection de changements dans des scènes complexes et la flexibilité des techniques de traitement d'image pour s'adapter à des conditions variables.

2 Premier algorithme

La première version de notre algorithme comportait un masque, la mise en évidence des différences entre les images et le traitement des bounding boxes. La première étape consistait à créer un masque permettant de différencier le sol du reste de la photo, afin de se focaliser directement sur le sol tout en ignorant les éventuels éléments parasites qui pourraient changer, comme les meubles ou autres. Ces masques ont été créés manuellement, avec un masque personnalisé pour chaque salle.

Ensuite, pour mettre en évidence les changements, nous avons calculé la différence entre l'image traitée et l'image de référence, toutes deux en niveaux de gris, puis appliqué un seuillage en utilisant la méthode d'Otsu. De plus, nous avons mis en œuvre une variante du traitement top-hat avec un noyau carré de 10x10. Enfin, nous avons identifié les contours en utilisant la bibliothèque OpenCV et obtenu des bounding boxes. Les résultats sont présentés dans la figure 1.

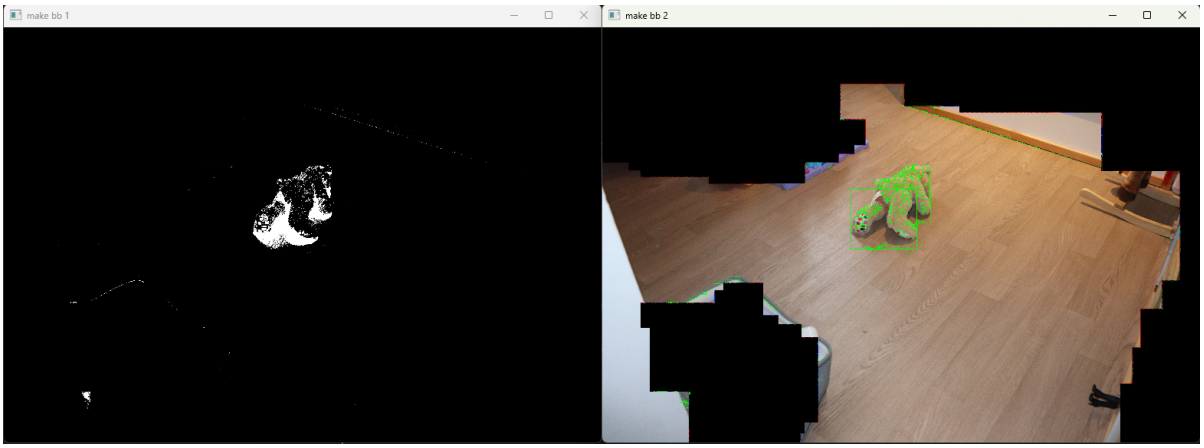

Figure 1 – Différences entre les grayscale + Otsu (gauche), Bounding Boxes (droite)

Pour améliorer la qualité des bounding boxes, nous avons effectué un traitement sur la liste des bounding boxes obtenues. La démarche consistait à supprimer toutes les

bounding boxes ayant une aire totale inférieure à $7000px^2$ et à fusionner les bounding boxes qui se chevauchent de plus de 10%. Cela nous a permis d'obtenir les résultats présentés dans la figure 2.

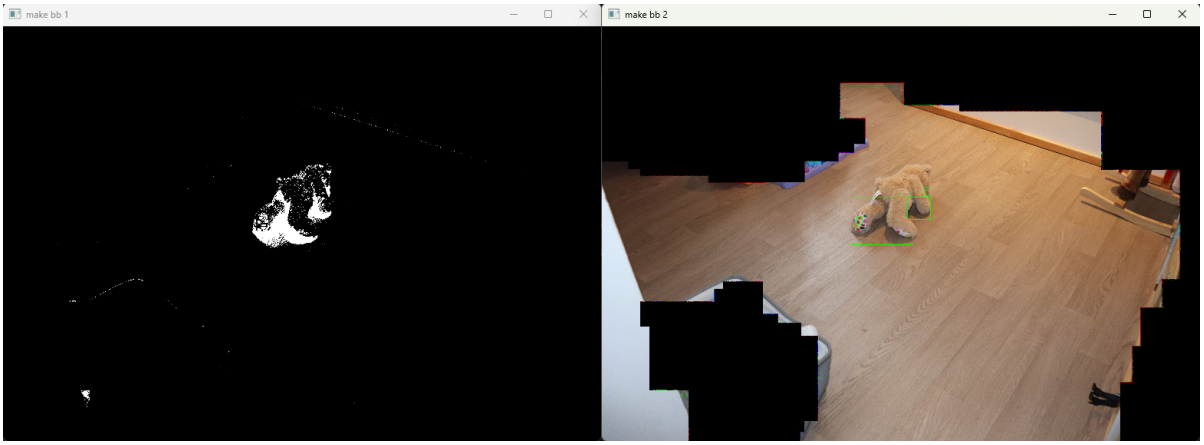

Figure 2 – Différences entre les grayscale + Otsu (gauche), Bounding Boxes (droite)

Cependant, les résultats ne sont pas aussi satisfaisants lorsque nous avons un changement de luminosité, comme le montre la figure 3.

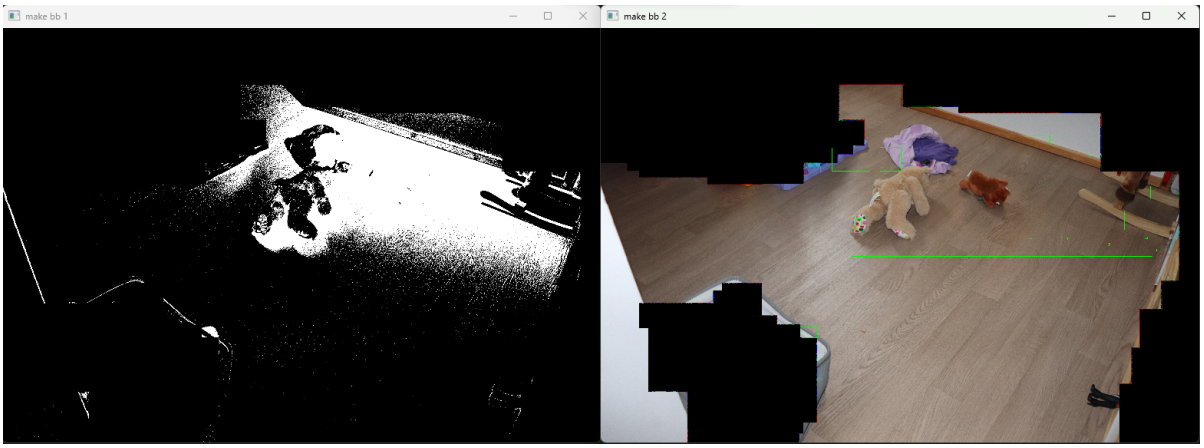

Figure 3 – Différences entre les grayscale + Otsu (gauche), Bounding Boxes (droite)

3 Problèmes de luminosité

Pour cette deuxième tentative, l'objectif était de résoudre le problème de luminosité. Notre première approche a consisté à équilibrer les histogrammes de manière à aligner l'histogramme de l'une des images sur l'autre. Les résultats de cette tentative sont visibles dans la figure 4. On remarque que les résultats sont à peine meilleurs que dans la version précédente.

La deuxième méthode d'équilibrage de la luminosité que nous avons essayée était d'utiliser l'espace CIELAB. Cependant, cela n'a pas été significativement plus efficace. Cependant, en travaillant avec cet espace et en observant certains résultats prometteurs, nous avons eu l'idée de tenter une approche différente.

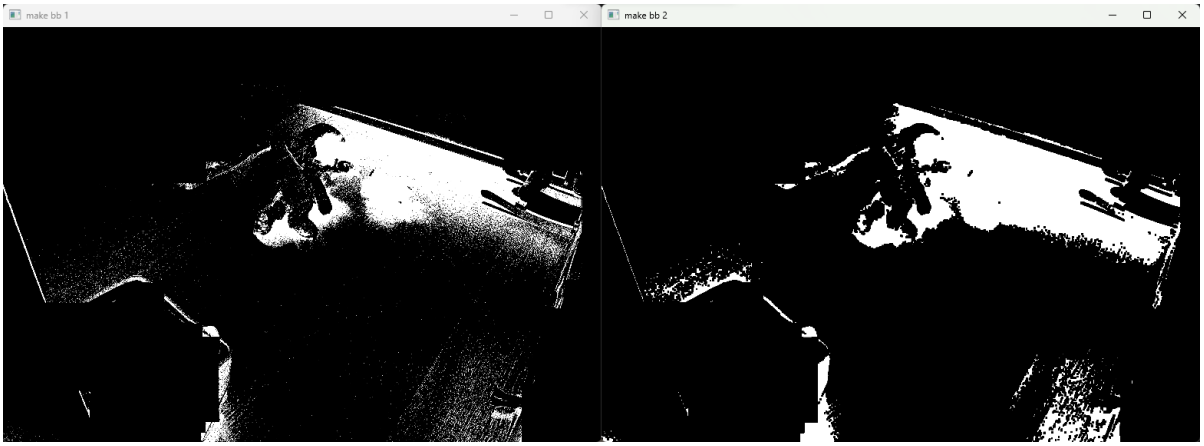

Figure 4 – Différences entre les grayscale + Otsu (gauche), Morphologie (droite)

4 Deuxième algorithme

Nous avons décidé d'essayer de comparer les contours des objets sur les deux images en effectuant une extraction de contours en premier lieu. Pour ce faire, nous avons réalisé une extraction de contour de manière naïve en appliquant un flou sur chaque image en niveaux de gris à l'aide d'un noyau carré de 100x100. En prenant la différence entre le niveau de gris de base et le niveau de gris flouté, nous obtenons les contours. Ensuite, nous avons extrait les contours de l'image que nous traitons et ceux de l'image de référence, puis appliqué un seuillage Otsu et une opération de morphologie de type top-hat. Cela nous a permis d'obtenir les résultats présentés dans la figure 5. Le workflow est illustré dans la figure 6.

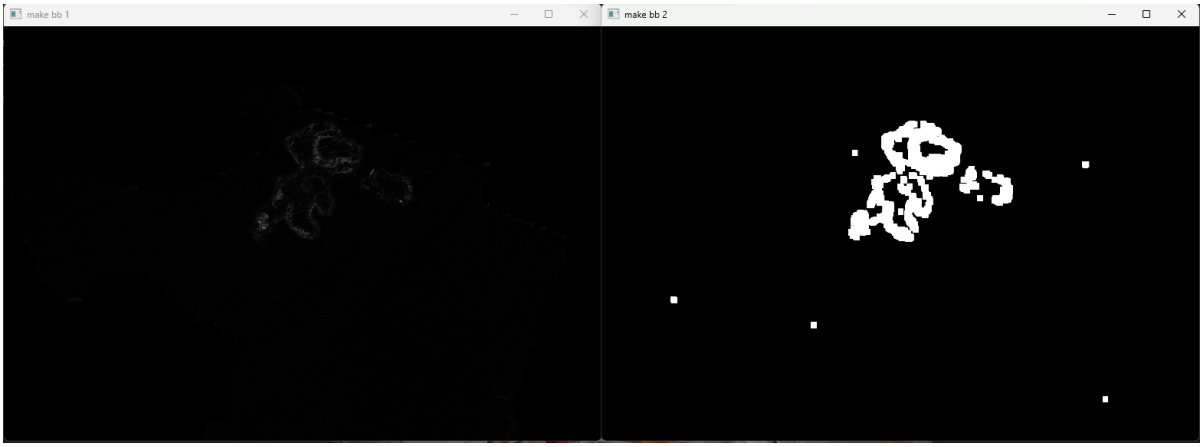

Figure 5 – Différences entre les contours (gauche), Morphologie (droite)

Cela nous permet alors d'extraire des bounding boxes de qualité, comme illustré dans la figure 7.

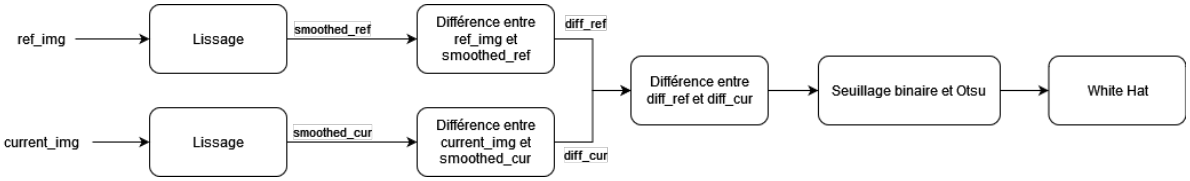

Figure 6 – Workflow pour extraire les contours des objets sur le sol

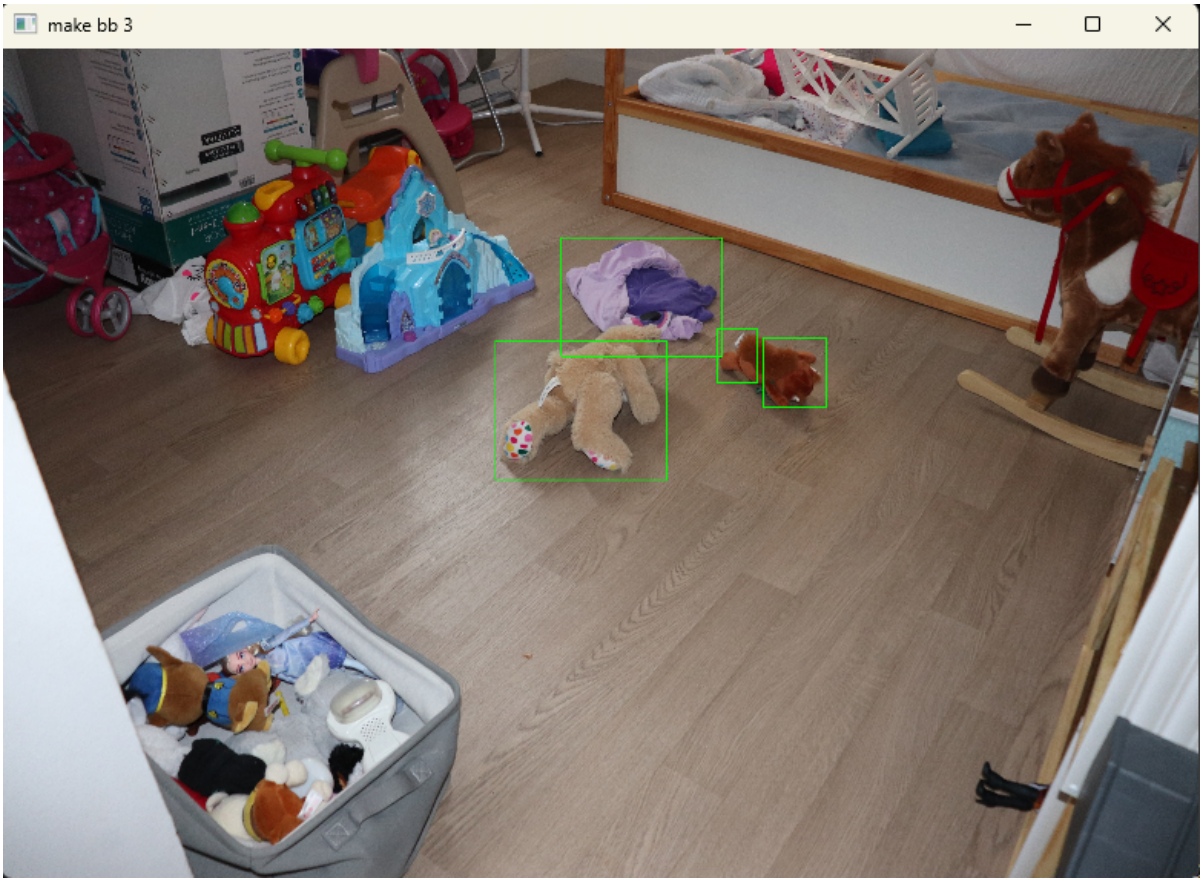

Figure 7 – Résultats final de notre algorithme sur une image de luminosité différentes

5 Code

Le code final peut être exécuté avec la commande *python TP1.py path-to-images*. Le fichier contenant les images devra contenir un dossier par pièce contenant au moins une image 'Reference.jpg' et une image 'Mask.JPG'. Le programme génère les images de sortie 'image_name_bb.jpg' avec les bounding boxes et les données d'encombrement superposées