Documentation Billard 2020

MERY Tom
LELIEVRE Maxime

1 Compatibilité

— Plateforme : Windows
— Compilateur : Visual Studio Code

2 Fichiers

— Documentation.pdf
— Billard2020.vi : VI principal à lancer en premier
— Call_C.vi : Sub-VI qui lance le programme C (Pix2Pos.exe) et renvoie les coordonnées des boules
— Draw_Line.vi : Sub-VI qui trace la trajectoire des boules dans Billard2020
— Get_Box.vi : Sub-VI qui détermine les coordonnées des bords du billard avec deux boucles imbriquées
— Get_Box_NO_Loop.vi : Sub-VI qui détermine les coordonnées des bords du billard sans boucle
— MP_LaunchMatlabScript.vi : Sub-VI qui lance le script Matlab Analyse.m
— Save_to_Matlab.vi : Sub-VI qui crée le script Matlab Analyse.m
— Pixmap.bin : Fichier binaire contenant les informations d'une image
— Pix2Pos.c : Code source de Pix2Pose.exe
— Pix2Pos.exe : Créer Pos.txt à partir de Pixmap.bin
— Pos.txt : Fichier texte contenant les coordonnées et le score de chaque boule
— Script.m : Sous-partie de Analyse.m
— Analyse.m : Script Matlab qui crée la feuille de résultat au format PDF (ScoreSheet_Fi.pdf)
— ScoreSheet_Fi.pdf : Feuille de résultat de la séquence i

3 Flow de données

- Billard2020.vi lance le sub-VI Get_Box.vi avec comme paramètres :
  — En entrée : le chemin de la 1ère image de la séquence et les ranges de couleurs pour le contour du billard.
  — En sortie : les coordonnées des bords du billard
- Billard2020.vi lance le sub-VI Call_C.vi de manière itérative : la boucle for lance Call_C.vi pour chaque image de la séquence avec comme paramètres :
  — En entrée : le chemin de chaque image de la séquence, les paramètres pour le code C (ranges des couleurs et diamètre des boules) et les coordonnées des bords du billard venant de Get_Box.vi.
  — En sortie : l'image du billard et les coordonnées des boules correspondant à l'image de la séquence.
  — Call_C.vi crée le fichier Pixmap.bin à partir du chemin de chaque image et assemble la ligne de commande à partir des paramètres d'entrée (paramètres sur les ranges de couleurs, le diamètre et les coordonnées des bords du billard). Puis Call_C.vi lance le VI System Exec.vi qui lance Pix2Pos.exe. Pix2Pos.exe lit le fichier binaire Pixmap.bin créé précédemment et génère le fichier Pos.txt où sont affichés les coordonnées et le score des boules pour une image. Call_C.vi lit le fichier Pos.txt et remplit un tableau avec les coordonnées des boules.
  — En cas d'erreur générée par Pix2Pos.exe, Call_C.vi modifie le cluster d'erreur en changeant le statut, le code d'erreur et affiche le message d'erreur disponible dans stderr.
- Billard2020.vi lance le sub-Vi Draw_Line.vi de manière itérative : la boucle for lance Draw_Line.vi (à la suite de Call_C.vi) pour chaque image de la séquence avec comme paramètres :
  — En entrée : les coordonnées des bords du billard, l'image du billard en sortie de Call_C.vi ainsi que les coordonnées des balles depuis le début de la séquence

  - En sortie : nouvelle image avec la trajectoire de chaque balle et les bords du billard dessiné en surimpression sur l'image de sortie de Call_C
  - L'image de sortie de Draw_Line.vi est imprimé dans la boucle d'appel et visionable dans l'interface utilisateur
- Après que la boucle for ait parcouru toutes les images de la séquence et si le booléen pour créer le fichier ScoreSheet_Fi.pdf a été sélectionné, Billard2020.vi lance le sub-VI Save_to_Matlab.vi qui va créer le script Matlab avec comme paramètres :
  - En entrée : le tableau regroupant toutes les coordonnées de la séquence, les paramètres d'affichage, le chemin de la 1ère image de la séquence (pour H et W) et la suite du script Matlab (script.m).
  - En sortie : le script complet sous forme d'un string .
- Le fichier Analyse.m est généré à partir du string de sortie de Save_to_Matlab à chaque éxcution de Billard2020.vi. Le VI MP_LaunchMatlabScript.vi lance le script Analyse.m dans Matlab qui va créer le fichier ScoreSheet_Fi.pdf dans le dossier courant.

Le cluster d'erreur d'entrée et de sortie est relié à chaque sub-VI afin de respecter l'ordre d'éxecution énoncé. L'arrêt spontanné de Billard2020.vi survient dès lors que le cluster d'erreur change de statut.

4 Erreurs :

Labview permet à l'utilisateur de définir des codes d'erreur personnalisés dans les gammes de -8999 à -8000, de 5000 à 9999 et de 500000 à 599999. Les erreurs générées par Pix2Pos.exe ont des codes comrpis entre 5000 et 5020.

Lorsque Pix2Pos.exe est appelé par Billard2020, les erreurs dans stderr sont redirigées dans le cluster d'erreur disponible dans l'interface utilisateur de Billard2020.

Pour chaque erreur survenant lors de l'éxecution de Pix2Pos.exe, l'espace mémoire alloué est libéré, les détails de l'erreur sont affichés dans stderr et le programme s'arrête avec une valeur de retour spécifique. A l'exception de l'erreur 5004, les erreurs suivantes provoquent l'arrêt spontané de Pix2Pos.exe et par extension de Billard2020 sans créer Pos.txt. L'erreur 5004 est quant à elle ingnoré par Labview et Pos.txt est créé.

Code	Description
1	'Pix2Pos' n'est pas reconnu en tant que commande interne ou externe, un programme exécutable ou un fichier de commandes. Cette erreur survient lorsque Pix2Pos.exe a été renommé ou supprimé du dossier courant
5001	Une ou plusieurs balles manquantes. Les balles manquantes sont indiquées dans stderr.
5002	Erreur lors de l'ouverture/création en écriture de Pos.txt.
5003	Erreur lors de l'écriture dans Pos.txt.
5004	Les boules se superposent. Les boules superposées sont indiquées dans stderr. Dans le cas où les trois boules se superposent avec au mois une autre boule le message d'erreur généré dans stderr est vrai mais incomplet.
5005	Erreur lors de la l'allocation de la mémoire.
5006	Erreur dans la ligne de commande. Pas le bon nombre de paramètres dans la ligne de commande et/ou diamètre de la boule en dehors de bornes [5..20]. Le cas est spécifié dans stderr.
5007	Erreur lors de l'ouverture en lecture de Pixmap.bin.
5008	Erreur lors de la lecture de Pixmap.bin.
5009	Largeur et/ou hauteur de l'image en dehors de bornes [10..1000].
5010	Pas assez de pixels. Le nombre de pixels manquants est indiqué dans stderr.
5011	Trop de pixels. La position des N pixels excédentaires n'étant pas identifiée, il est risqué de choisir d'ignorer les N derniers pixels de Pixmap.bin. Le programme s'arrête.
5012	Erreur lors de la fermeture de Pixmap.bin
5013	Erreur lors de la fermerture de Pos.txt.
5021	Le chemin d'accès de la séquence fournie dans l'interface utilisateur de Billard2020.vi n'existe pas.
5022	Le chemin d'accès de la séquence fournie dans l'interface utilisateur de Billard2020.vi ne contient aucun fichier png.

5 Algorithms

5.1 Get_Box.vi

Recherche les bords du billard avec deux boucles imbriquées.

Entrées :
- Matrice T de taille WxH de unsigned int 32-bit contenant chaque pixel de l'image au format 0x00RRGGBB
- Range de couleurs du bords du billard : R_{min}, R_{max}, G_{min}, G_{max}, B_{min}, B_{max}
- Largeur W, Hauteur H

Initialisation des coordonnées du billard : X_{min}=Largeur ; X_{max}=0 ; Y_{min}=Hauteur ; Y_{max}=0

- Pour i allant de 1 à Hauteur (parcours toutes les lignes)
  - Pour j allant de 1 à Largeur (parcours toures les colonnes)
    - Si le pixel courant Tij appartient aux ranges de couleurs du bords du billard :
      - Si $X(T_{ij}) < X_{min}$ alors $X_{min} = X(T_{ij})$
      - Si $X(T_{ij}) > X_{max}$ alors $X_{max} = X(T_{ij})$
      - Si $Y(T_{ij}) < Y_{min}$ alors $Y_{min} = Y(T_{ij})$
      - Si $Y(T_{ij}) < Y_{max}$ alors $Y_{max} = X(T_{ij})$

Sorties : X_{min} ; X_{max} ; Y_{min} ; Y_{max}

5.2 Get_Box_NO_Loop.vi

Recherche les bords du billard par vectorisation (sans boucle).

Entrées :
- Matrice T de taille WxH de unsigned int 32-bit contenant chaque pixel de l'image au format 0x00RRGGBB
- Range de couleurs du bord du billard : R_{min}, R_{max}, G_{min}, G_{max}, B_{min}, B_{max}
- Largeur W, Hauteur H

Isole les composantes R, G et B de chaque éléments de T en trois matrices T_R, T_G et T_B de taille WxH en divisant les unsigned int 32-bit de T en trois unsigned int 8-bit.

Converti T en matrice logique contenant des 1 aux index où les pixels appartiennent aux ranges de couleurs du bords du billard :
- $T_R = R_{min} \leqslant T_R \leqslant R_{max}$
- $T_G = G_{min} \leqslant T_G \leqslant G_{max}$
- $T_B = B_{min} \leqslant T_B \leqslant B_{max}$
- $T = T_R$ & T_G & T_B & : oprérateur logique "ET"

Converti T logique en vecteur 1D : $A(Wi + j) = T_{ij}$ $i, j \in N : i \in [0, H-1]$ et $j \in [0, W-1]$
Converti T^T logique en vecteur 1D : $B(Hj + i) = T_{ji}$ $i, j \in N : i \in [0, H-1]$ et $j \in [0, W-1]$
- I_A=index du premier élément de A égale à 1
- I_B=index du premier élément de B égale à 1
- I_C=index du dernier élément de A égale à 1
- I_D=index du dernier élément de B égale à 1

Les bords du billard sont ensuite définie comme :
- Y_{min} est le quotient de la division euclidienne de I_A par W avec un reste défini positif
- X_{min} est le quotient de la division euclidienne de I_B par H avec un reste défini positif
- $Y_{max} = H - Q_C$ avec Q_C le quotient de la division euclidienne de I_C par W avec un reste défini positif
- $X_{max} = W - Q_D$ avec Q_D le quotient de la division euclidienne de I_D par H avec un reste défini positif

Sorties : X_{min} ; X_{max} ; Y_{min} ; Y_{max}

5.3 Pix2Pos.exe (fonction Ball_Search dans code source)

Recherche les coordonées des balles avec quatres boucles imbriquées. L'algorithm parcourt tous les carrés de 11x11 pixels à l'intérieur des bords du billard, identifie celui avec le plus grand score (nombre de pixel appartenant aux ranges de couleurs d'une balle donnée) et retourne les coordonnées Top-Left du carré avec le plus grand score pour chaque balle.

Entrées :
- Largeur W, Hauteur H
- Bords du billards : X_{min} ; X_{max} ; Y_{min} ; Y_{max}
- Tableau de struct Pixel (voir definition dans code source Pix2Pos.c) de longueur WxH

Initialisation des scores max de chaque balle et création d'une variable b de type struct Ball (voir definition dans code source Pix2Pos.c) afin de stocker les coordonnées Top-Left du carré de 11x11 avec le score max courant pour chaque balle :
- Score_Red=0 ; Score_Yellow=0 ; Score_White=0

- Pour i allant de Y_{min} à $Y_{max} - 11$ (parcourt les lignes à l'intérieur des bords du billard)
  - Pour j allant de X_{min} à X_{max-11} (parcourt les colonnes à l'intérieur des bords du billard)
    - Initialisation des scores du carré 11x11 de coordonnées $[j, i]$: Red=0 ; Yellow=0 ; White =0
    - Optimisation : Si le pixel du centre du carré 11x11 de coordonées $[j, i]$ appartient au range de couleurs du fond bleu du billard alors on passe directement au carré suivant (j=j+1)
      - Pour k allant de 0 à 10 (parcourt les lignes du carré 11x11 de coordonnées $[j, i]$)
        - Pour l allant de 0 à 10 (parcourt les colonnes du carré 11x11 de coordonnées $[j, i]$)
        - Red=somme des pixels du carré 11x11 de coordonnées $[j, i]$ appartenant aux ranges de couleurs de la balle rouge (information pour chaque pixel contenu dans variable de type struct Pixel)
        - Yellow=somme des pixels du carré 11x11 de coordonnées $[j, i]$ appartenant aux ranges de couleurs de la balle jaune
        - White=somme des pixels du carré 11x11 de coordonnées $[j, i]$ appartenant aux ranges de couleurs de la balle blanche
    - Si Score_Red<Red alors :
      - Score_Red=Red ;
      - Les coordonnées [j,i] de la balles rouge sont stockées dans b
    - Si Score_Yellow<Yellow alors :
      - Score_Yellow=Yellow ;
      - Les coordonnées [j,i] de la balle jaune sont stockées dans b
    - Si Score_White<White alors :
      - Score_White=White ;
      - Les coordonnées [j,i] de la balles sont stockées dans b

Sorties :
- Score_Red ; Score_Yellow ; Score_White
- Varibale b de type struct Balls

Les sorties de cet algorithm de recherche sont ensuite imprimées dans Pos.txt.

5.4 Analyse.m

Entrées :
- Coordonnées successives des balles
- Paramètres de personalisation de ScoreSheet_Fi.pdf à définir dans l'interface utilisateur de Billard2020.vi

Sortie :
- Création de ScoreSheet_Fi.pdf

Pour étudier la trajectoire de chaque balle, un vecteur "déplacement" D de taille 3xn (n le nombre d'image de la séquence) est créé. Les lignes 1, 2 et 3 sont dédiées respectivement aux mouvements des balles rouge, jaune et

blanche. Les valeurs de D sont définies dans le plan complexe comme :

$$\begin{cases} Re\{D(1,:)\} = diff(X_{Red}) & Im\{D(1,:)\} = diff(Y_{Red}) \\ Re\{D(2,:)\} = diff(X_{Yellow}) & Im\{D(2,:)\} = diff(Y_{Yellow}) \\ Re\{D(3,:)\} = diff(X_{White}) & Im\{D(3,:)\} = diff(Y_{White}) \end{cases}$$

Nous faisons l'hypothése qu'une balle autre que la balle du joueur est en mouvement si et seulement si elle a été touchée par la balle du joueur et nous supposons que les balles autre de celle du joueur ne se percutent pas entre elles.

5.4.1 Détection du mouvement des balles

La balle i est considérée en mouvement lorsque :

$$\|D(i,j)\| > dmin \qquad \text{avec } dmin = \sqrt{2} \ ; \ i = 1,2,3 \ et \ j \in [1,n] \in N$$

5.4.2 Détection des rebonds

La plan complexe étant divisé en 4 quadrants, trouver les points où les balles percutent une bande revient à trouver les indexes dans D où les points complexes successifs changent de quadrant et sont proches du bord du billard. Pour cela on applique à D les conditions logiques :

$$[sign(Re\{D(i,j)\}) \neq sign(Re\{D(i,j+1)\})] \ OU \ [sign(Im\{D(i,j)\}) \neq sign(Im\{D(i,j+1)\})]$$

$$avec \ i = 1,2,3 \ et \ j \in [1,n] \in N$$

Il faut rajouter à cela une condition qui vérifie que la position de la balle est proche du bord du billard. De plus la valeur absolue de la différence d'argument de deux valeurs de D succesives doit être supérieure à un certain seuil pour ne pas prendre en compte les changements de signe dû aux bruits de la séquence lors de déplacements horizontaux ou verticaux.
Ainsi les index vérifiant ces conditions sont stockés dans un vecteur logique (BdT dans Analyse.m) de taille 3xn et il suffit d'appliquer la bonne ligne au bon couple de vecteur X et Y pour afficher les rebonds de la balle du joueur sur ScoreSheet_Fi.pdf.
Dans le cas où une des balles n'est pas touché le nombre de rebonds calculés et affichés dans ScoreSheet_Fi.pdf est 0.
ScoreSheet_Fi.pdf affiche uniqument le premier choc avec la balle du joueur.

5.4.3 Détection des outliers

Les indices des outliers sont calculés avec la fonction isoutlier sur les distance à l'origine ($\sqrt{X^2 + Y^2}$) et la condition :

$$\|D(i,j)\| > dmax \qquad \text{avec } dmax = 200 \ ; \ i = 1,2,3 \ et \ j \in [1,n] \in N$$

doit également être vérifié pour ne pas trouver d'outlier sur les séquences valides.