



UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

Reconnaissance de boîtes de jeux de plateaux

Rapport de IN52 - IN54

BOUCHEREAU Thomas - BRUNET Pierre - PERREZ Stéphane

Département Informatique - Imagerie, Interaction et Réalité Virtuelle

Université de Technologie de Belfort Montbéliard
90 010 - BELFORT CEDEX

Responsable IN52
Cindy CAPPELLE

Responsable IN54
Yassine RUICHEK

Table des matières

1	Objectifs du projet	3
2	Développement	3
2.1	Récupérer la face du dessus de la boîte de jeu	3
2.2	De la 3D à la 2D	5
2.3	Analyse des fréquences	6
2.4	Analyse des résultats	7
2.5	Interface graphique	7
2.6	Mode d'emploi	7
3	Conclusion	7

1/ Objectifs du projet

Le but du projet est de faire de la reconnaissance de jeux de plateaux à partir de leur boîte de jeu et d'une base de données. Il nous faut donc procéder en plusieurs étapes. Ainsi, nous devons recentrer la photo sur la face du dessus puis la recadrer afin d'avoir la meilleure analyse possible. Ensuite, nous allons comparer le pourcentage de couleur entre les jeux de la base de données et la photo. Cette base de données est censée se créer au fur et à mesure à l'aide d'une interface.

Nous avons décidé de diviser le projet en trois parties. Ainsi, le premier étudiant s'occuperait du recadrage de la face du dessus de la boîte. Le second prendrait en charge la comparaison de deux images données. Enfin, le dernier créerait la base de données et l'interface graphique afin que l'application soit utilisable en dehors de tests divers.

2/ Développement

2.1/ Récupérer la face du dessus de la boîte de jeu

Pour récupérer la face du dessus, il est nécessaire de pouvoir distinguer les contours dans l'image. Nous avons donc décidé d'utiliser la transformée de Hough pour cela. Mais avant tout, il nous faut enlever le bruit de l'image afin d'éviter des contours inutiles. Nous avons donc utiliser la technique d'érosion puis avons rogné l'image en fonction de la densité restante dans les lignes de l'image.



FIGURE 1 – Image binaire et érodée



FIGURE 2 – Image binaire rognée

Après ce pré-traitement, on applique la transformée de Hough à des lignes faisant au moins une certaine longueur pour éviter d'avoir des lignes du dessin du jeu. On allonge ensuite ces lignes et on récupère les points d'intersection.

Ces points d'intersection ont été produits grâce aux fonctions *polynomes* et *polyval*. Cependant, les lignes générées ne sont pas forcément dans le bon ordre. Par conséquent, les points ne le sont pas non plus. De plus, on génère toutes les intersections de lignes possibles. Il faut donc faire un tri pour éviter les duplicas et également les trier dans un ordre donné. C'est le dernier traitement appliqué avant de passer à l'analyse de l'image.

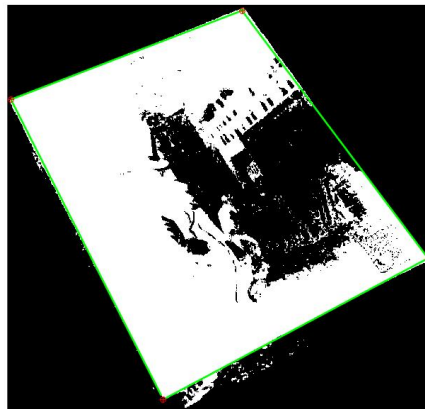


FIGURE 3 – Contours de la face du dessus

2.2/ De la 3D à la 2D

Une fois les points récupérés, il nous faut adapter l'image pour la rendre comparable avec l'image de base. On utilise donc les points d'intersection précédemment calculés.

La fonction `fitgeotrans` permet de rapprocher une image d'une autre grâce à un tableau de points mouvants et un tableau de points fixes. Le tableau de points mouvants doit *rentrer dans* (*fit* en Anglais) le tableau de points fixes.

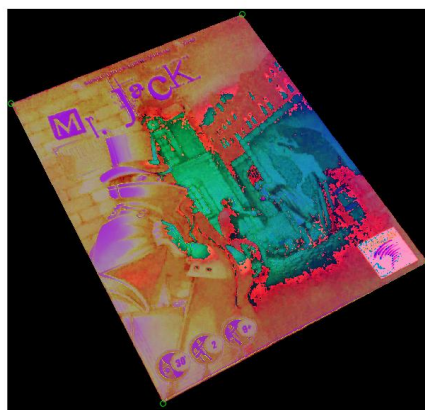


FIGURE 4 – Points mouvants en vert sur l'image

C'est à partir de ce moment-là que nous avons rencontré des problèmes. En effet, la fonction n'est pas aussi efficace qu'elle le sous-entend. Malgré tous nos efforts, nous n'avons pu que redresser la boîte sans pouvoir changer les angles de l'image.

Vous verrez sur les figures 5 et 6 les résultats obtenus grâce à la fonction Matlab `fitgeotrans` et à un recadrage.

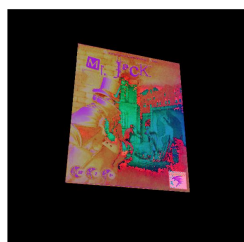


FIGURE 5 – Résultat de la fonction *fitgeotrans*



FIGURE 6 – Recadrage maximal obtenu

2.3/ Analyse des fréquences

Nous avons donc décidé de revenir à l'image pré-traitée et ne comparer que les fréquences de couleurs afin de pouvoir rapidement reconnaître un jeu. Cela s'est révélé plus logique et plus optimisé en prenant comme couleurs le HSL. Nous n'avons alors qu'à récupérer le Hue (à savoir la nuance) de chaque pixel et comparer les densités par rapport à la base de données.

Pour savoir quel photo correspond à quel jeu, nous avons récupéré les dessus des boîtes de jeux (trouvés sur des sites marchands) et nous leur avons associé le nom du jeu dans une base de données.

Une fois la photo transformée et recadrée, sans la transformation de *fitgeotrans* pour gagner du temps, nous la comparons avec la base de données.

Voici l'exemple du jeu Loony Quest et de son analyse d'histogramme :

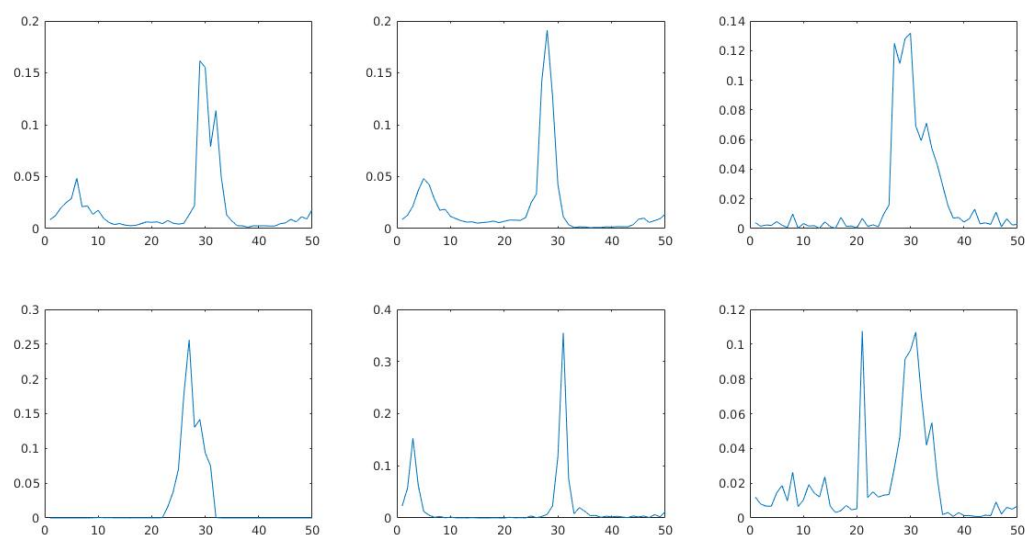


FIGURE 7 – Histogrammes de la photo Loony Quest recadrée et des 5 plus proches voisins

On voit ainsi que les deux premiers histogrammes sont relativement proches mais peu pratiques à comparer pour un ordinateur. Nous avons donc décidé de créer une fonction calculant la ressemblance entre deux histogrammes. Elle utilise la fonction Matlab *corrcoef* mais avec un delta permettant de décaler la courbe et ainsi vérifier que les couleurs sont proches ou non.

2.4/ Analyse des résultats

Sur 10 photos de jeux différents testés, 6 fonctionnent sans soucis. Les 10 sont ainsi bien recadrés et analysés mais certains sont peu pratiques à l'analyse. Ainsi, le Takenoko et le IKnow ne fonctionnent pas en raison d'une boîte plus claire que les autres et donc plus sensible au peu de luminosité qu'il y avait lors de la prise de photo.

En revanche, les 6 photos qui fonctionnent ont un pourcentage de ressemblance avoisinant les 90% pour la plupart. Nous pensons donc que la base de données pourra être complétée à l'avenir et évoluer sur un apprentissage poussé tel que le deep learning.

2.5/ Interface graphique

Pour permettre à l'utilisateur de facilement utiliser le programme, nous avons mis en place une interface permettant de visualiser le résultat le plus proche ainsi qu'un tableau des cinq meilleures estimations quant à l'identité du jeu.

2.6/ Mode d'emploi

Pour lancer l'application, il suffit de lancer :

```
$ ./run_gui.sh 'PATH_OF_R2016A_FOLDER'
```

Vous pouvez ensuite tester différentes images en mettant le chemin relatif dans le champ de texte demandé et en appuyant sur Entrée. Par exemple, pour l'image de Abyss, il faut préciser "Images/abyss.jpg".

3/ Conclusion

Ce projet a été pour nous très utile et intéressant. En effet, nous nous sommes rendus compte de la difficulté de traitement de photos et nous avons pu tenter plusieurs méthodes afin de retrouver le jeu correspondant. Nous avons ainsi pu utiliser des outils encore peu connus de nous-mêmes sur Matlab. De plus, nous avons appris de nouvelles méthodes de comparaison telles que celle du Coefficient de Corrélation utilisée dans notre programme. Enfin, nous avons pu tenter en conditions les éléments appris en cours et en travaux pratiques et dirigés, cela permet d'avoir une expérience réelle de l'imagerie que ce soit dans le traitement comme dans l'analyse d'une image. Nous pensons avoir pu concilier les deux matières dans ce projet.

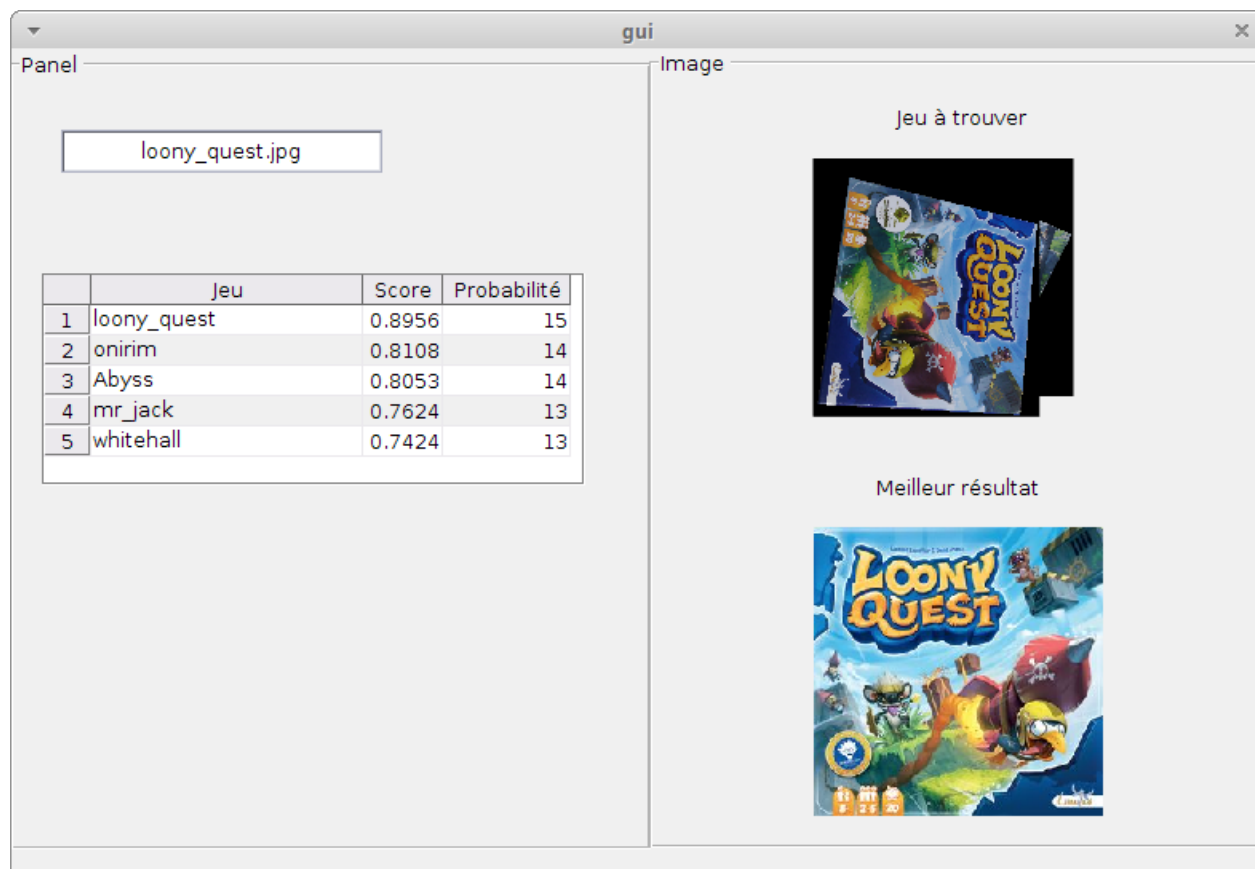


FIGURE 8 – Histogrammes de la photo Loony Quest recadrée et des 5 plus proches voisins

Dans les améliorations possibles, nous pensons qu'il serait intéressant de combiner plusieurs classifieurs mais également d'intégrer un algorithme de deep-learning. De plus, après avoir utilisé les outils disponibles de Matlab, nous nous demandons s'il ne serait pas plus viable de demander à l'utilisateur de centrer la photo de lui-même. Cela éviterait ainsi du temps de calcul et permettrait d'être encore plus précis dans la comparaison.