



Manuel d'utilisation

Présentation

hxlris est un logiciel de reconnaissance vidéo permettant le développement d'applications basé sur les gestes ou la position de l'utilisateur.

Le logiciel base sa reconnaissance sur des zones de couleurs, pouvant très rapidement être mis en œuvre, par exemple avec des pastilles de couleurs collés aux endroits dont la reconnaissance est nécessaire.

Avec ce manuel se trouve les sources ainsi qu'une version compilée du logiciel comprenant deux exemples afin d'en tester l'utilisation.

Cahier des charges

Objectifs et contraintes

Au cœur du logiciel, la reconnaissance vidéo nécessite d'être rapide et précise afin d'être utilisée dans une application temps réel (au moins 30 images par secondes) et pouvoir détecter les caractéristiques suivantes :

- Angle d'un objet
- Distance d'un objet par rapport à la caméra
- Position d'un objet dans le champ de détection de la caméra

Grâce à ces trois informations, on peut établir un ensemble de schéma d'interaction possibles avec l'utilisateur. En termes de contraintes, cela nécessite pour l'utilisateur d'avoir l'objet reconnu face à la camera.

Un contrainte que j'ai estimé nécessaire à mon projet est également le caractère multi-plateforme du logiciel. En effet, pour que ce soit utilisable par une grande proportions d'applications, j'ai estimé nécessaire que hxlris fonctionne aussi sur PC ou Mac que sur Smartphone ou dans un navigateur Web. Cette contrainte m'a empêché de me tourner vers des techniques à matériel dédié très utilisés aujourd'hui comme la technologie Kinect de Microsoft car celle-ci n'est pas compatible sur smartphone par exemple.

Bibliothèques utilisées

Pour développer hxlrís, j'ai choisi le langage Haxe. C'est un langage moderne qui devient de plus en plus utilisé au fur et à mesure de son développement. Les principaux avantages de ce langage et du compilateur livré avec son une syntaxe moderne et orienté objet ainsi qu'une compilation vers différentes plateformes (PC, Mac, Smartphone, Serveur, Web). Ces différents avantages ainsi que le dynamisme de sa communauté et la relative jeunesse de ce langage m'a orienté vers ce choix.

Puisque le logiciel comprend une large composante nécessitant de l'analyse vidéo, j'aurais pu utiliser OpenCV pour effectuer les traitements nécessaires à ce sujet. La popularité et la documentation bien fournie de cette bibliothèque en aurait fait une base solide pour hxlrís.

Cependant, les exemples que j'ai pu voir avec OpenCV ne me laissaient pas supposer d'un degré de précision suffisant lors de la reconnaissance. Cela peut se comprendre par le fait qu'OpenCV soit orienté sur la reconnaissance de zone. J'ai donc choisi d'implémenter moi-même les fonctionnalités d'analyse vidéo.

J'ai choisi d'utiliser les bibliothèques :

- hxOpenFrameworks pour avoir accès au matériel, notamment la webcam, permettant ainsi de faire la reconnaissance vidéo.
- NME pour afficher des informations et des images à l'écran, c'est un framework largement utilisé dans le milieu du développement de jeu vidéo.
- Actuate pour réaliser quelques animations grâce aux fonctions d'interpolation incluses dans cette bibliothèque.

Fonctionnement

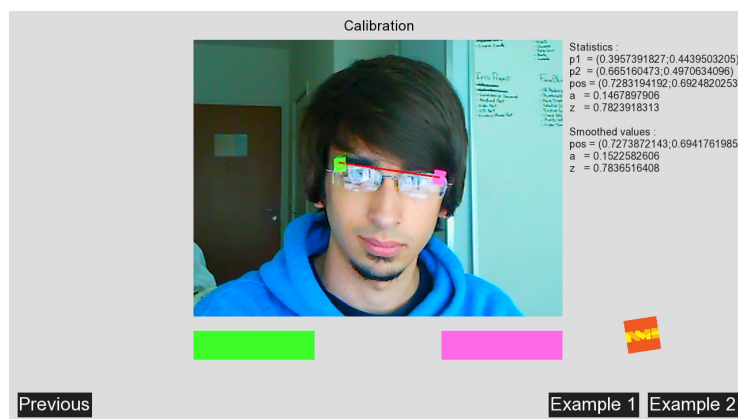
Pour la reconnaissance, le logiciel reconnaît pour chaque image les pixels correspondant à une certaine couleur jusqu'à un certain seuil. Il détermine ensuite la position globale de la zone de couleur. Avec deux zones de couleurs reconnues, on peut déterminer l'angle de l'objet correspondant à l'angle entre les deux zones, ainsi que la position par rapport à la camera. Enfin la profondeur est déterminée par rapport à la distance entre les deux zones.

Les valeurs ainsi obtenues sont ensuite lissées (grâce à une méthode de lissage par moyenne) permettant ainsi d'éviter le bruit de reconnaissance.

Le logiciel se décompose en quatre écrans parmi lesquels on peut naviguer grâce aux boutons présents dans l'interface, ou revenir en arrière grâce à la touche [Esc].

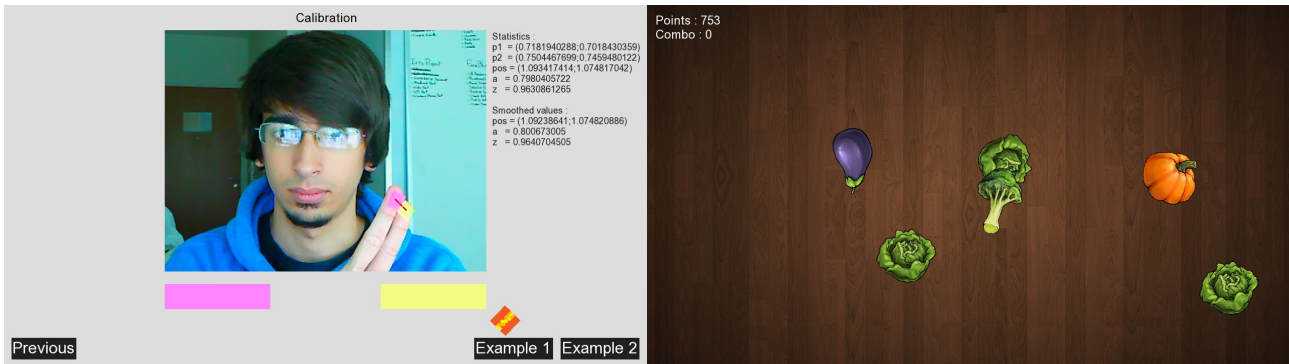
La premier écran est un écran de présentation montrant le logo du logiciel. Un bouton [Next] est présent pour naviguer vers l'écran de Calibration.

L'écran de calibration montre une image de la capture vidéo réalisée ainsi que les deux couleurs actuellement reconnues par le logiciel. Ces couleurs sont paramétrables l'une après l'autre en cliquant sur la capture vidéo. Des statistiques sont affichées à droites, comme la position, l'angle et la profondeur de l'objet reconnu par rapport à la camera. Enfin, une image est présente en bas à droit de l'écran permettant de tester la reconnaissance de l'objet, cette image bouge et tourne avec l'objet. Deux boutons [Exemple 1] et [Exemple 2] permettent de naviguer vers les deux exemples fournis avec le logiciel.



Exemple 1

Le premier exemple est un jeu qui consiste pour l'utilisateur à placer le viseur sur des légumes qui apparaissent à l'écran, lui permettant ainsi d'augmenter son score. Il est recommandé de placer les pastilles de reconnaissance sur ses doigts pour que le viseur à l'écran se déplace avec les doigts de l'utilisateur.



Exemple 2

Le deuxième exemple affiche à l'écran plusieurs rectangles de couleur superposés. Il est recommandé pour cet exemple de placer les repères de couleurs sur sa tête. Les rectangles sont virtuellement séparés en profondeur, ce qui permet de les différencier en bougeant la tête et ainsi d'avoir une impression de profondeur dans l'écran.



Développements à venir

Le logiciel n'est pas encore disponible sur toutes les plateformes auquel il se destine. Un portage sur de nombreux dispositifs est donc prévu comme les systèmes Android, iOS, Windows Phone et les navigateurs Web.

Les résultats actuels sont dépendants des conditions de luminosité et de qualité de la caméra. Différents algorithmes pourront être implémentés pour résoudre ces problèmes. Une combinaison avec OpenCV ou différents types de reconnaissances sont envisageables.

Conclusion

Ce projet m'a permis d'expérimenter la reconnaissance vidéo dans un projet concret. Cela a également été une bonne occasion pour moi de mettre en pratique mes connaissances autour d'une technologie qui m'intéresse particulièrement. J'ai aussi pu constater les limites des méthodes que j'ai employé.