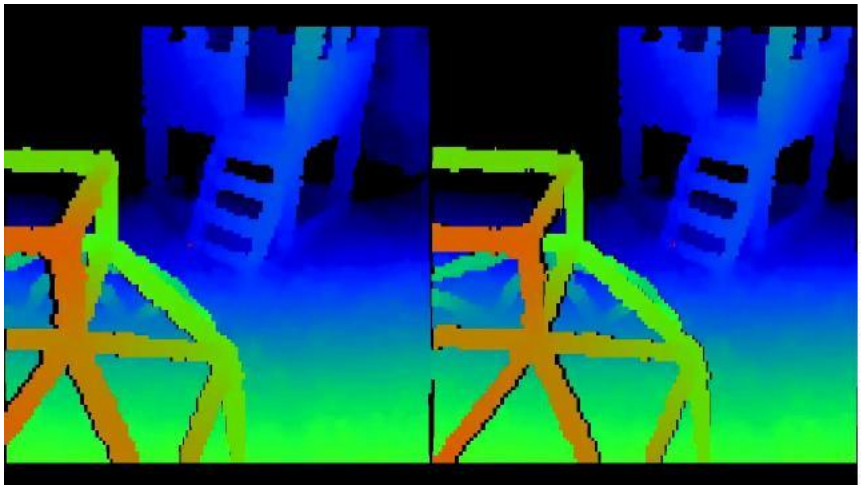
Voir le monde à travers un capteur de profondeur

Dans ce projet nous installons une caméra à capteur de profondeur sur un casque stéréoscopique (en résumé : Kinect + Google Cardboard). Il en résulte un appareil portable qui offre aux utilisateurs une vision surprenante de leur environnement.

http://www.pabr.org/depthvision



http://www.pabr.org/depthvision

READ THE HYPERTEXT VERSION HERE:
http://www.pabr.org/depthvision/depthvision.fr.html

Historique des versions

1.0	2014-11-10	Première publication.

Table des matières

1. Introduction

La réalité virtuelle arrive-t-elle enfin à maturité, après plusieurs décennies d'espoirs déçus ? Il est encore trop tôt pour le dire, mais il faut reconnaître qu'il y a eu des avancées significatives depuis notre dernière incursion dans ce domaine en 2009 avec [WXHMD] : Kinect, Google Glass, Oculus Rift, Google Cardboard, etc.

Dans ce projet nous combinons deux technologies abordables depuis peu : les caméras à capteur de profondeur, et les casques vidéo à base de smartphones. Une caméra à capteur de profondeur équivalente au Kinect est connectée au port USB OTG d'un téléphone Android. L'information de profondeur est affichée en stéréoscopie avec une latence très faible. Plusieurs modes de visualisations sont envisageables :

- carte de profondeur en fausses couleurs ;
- perspective amplifiée (par exemple comme si les yeux étaient espacés de 50 cm) ;
- vue en perspective à la troisième personne (projection depuis un point de vue arbitraire) ;
- vue de dessus de type "radar" ou "sonar".

D'autres projets ont déjà utilisé des caméras montées sur des casques vidéo, le tout connecté à des ordinateurs rapides, pour des applications sophistiquées de modélisation 3D et de suivi de mouvement. Ici nous privilégions le faible coût, la portabilité et la liberté de mouvement et nous nous contentons de fournir une expérience divertissante à l'utilisateur. Il s'agit de réalité "alternative" plutôt que de réalité "virtuelle" ou "augmentée".

2. Hardware

2.1. Configuration testée

Ce projet est possible parce que les smartphones sont maintenant suffisamment puissants pour servir d'hôte USB à des périphériques conçus pour les PCs de bureau. Notre réalisation se compose de :

- ASUS Xtion Pro Live: une caméra à capteur de profondeur très semblable au Kinect de la Xbox 360 (technologie PrimeSense à projection de motif IR) ;
- Samsung Galaxy S3: un smartphone Android relativement ancien (début 2012) avec un écran AMOLED de 1280x720 et un port USB OTG 2.0 ;
- Micro-B USB OTG cable: La broche ID indique au téléphone qu'il doit se comporter comme un hôte USB plutôt que comme un périphérique.
- VR-Spektiv XL: une variante de Google Cardboard en mousse, avec un écart inter-lentilles ajustable.

2.2. Alternatives

2.2.1. Caméra à capteur de profondeur

Nous avons choisi le ASUS Xtion (plutôt que le Kinect) pour sa taille réduite et sa connectique USB standard. Il est surprenant que le Galaxy S3 soit capable de l'alimenter, car les ports USB OTG sont généralement conçus pour des périphériques à faible consommation tels que les disques USB flash. Le Xtion consomme 300 mA dans notre application (transfert de données de profondeur à 160x120x30im/s, pas de traitements audio, pas de flux vidéo RVB).

N.B. : D'après certains témoignages l'ASUS Xtion ne serait pas compatible avec les ports USB 3.0, et il faudrait les configurer en mode 2.0 en chargeant le driver ehci sous Linux. Malheureusement on ne peut pas en faire autant aussi facilement sur un appareil Android.

Le Kinect de la Xbox 360 est évidemment l'alternative la plus naturelle. On devrait le trouver de plus en plus facilement sur le marché de l'occasion dans les prochaines années au fur et à mesure que des consoles plus récentes envahissent le marché. Notons qu'il a un connecteur propriétaire qui ajoute une entrée 12 V aux signaux USB. Un Kinect acheté séparément comme accessoire pour une Xbox 360 est semble-t-il livré avec une alimentation externe qui sert également d'adaptateur vers une connectique USB ordinaire. Le descripteur USB indique une consommation maximale de 100 mA sur le VBUS USB. Il devrait donc être possible de brancher un Kinect sur un port USB OTG 2.0 faible puissance.

Le Kinect v2 (pour la Xbox One) est évidemment très intéressant, mais il exige un port USB 3.0. Il semble possible d'insérer une fiche USB 3.0 Standard-B dans la prise au dos du capteur, mais il faut encore ensuite amener une alimentation 12 V à l'intérieur en réalisant des soudures.

Il existe aussi plusieurs modèles de capteurs de profondeur à courte portée, conçus avant tout pour détecter les mouvements des mains. Notons que contrairement au Kinect, certains de ces capteurs ne réalisent pas les traitements de corrélation stéréo en interne.

2.2.2. Alimentation externe

Pour améliorer l'autonomie, il est possible d'alimenter la caméra via un *Accessory Charging Adapter* USB.

Idéalement l'ACA devrait aussi alimenter l'appareil Android. Ce cordon que nous avons testé raccorde toutes les broches VBUS ensemble (fiche micro-B pour l'hôte OTG, prise Type-A pour le périphérique, fiche Type-A pour l'alimentation) mais le Galaxy S3 n'en profite pas pour se recharger. La broche ID ne présente apparemment la valeur de résistance spéciale qui déclenche la fonction ACA.

2.2.3. Appareil Android

Le Galaxy S3 donne satisfaction, mais avec une résolution de seulement 1280x720 on distingue facilement les pixels, et par ailleurs l'écran relativement petit restreint le champ de vision.

Si le coût n'était pas un facteur limitant, nous utiliserions un Samsung Galaxy Note 4 (écran AMOLED 2560x1440, 515 pixels par pouce, USB 3.0). La largeur de son écran est exactement le double de l'écart inter-pupillaire moyen, ce qui permet d'obtenir un champ de vision symétrique avec un casque de type Cardboard.

Le Samsung Galaxy Note 3 a également un écran de dimensions idéales, mais avec une résolution plus faible (1920x1080, 386 pixels par pouce).

Le Google Nexus 6 aura un écran légèrement plus grand.

Parmi tous ces appareils, seul le Galaxy Note 3 a un port USB 3.0 (vraisemblablement requis pour le Kinect v2).

N.B. : En l'état actuel le logiciel exige un appareil rooté pour accéder directement aux périphériques USB.

2.2.4. Casque stéréoscopique

Google Cardboard a popularisé le concept de casque de réalité virtuelle à base de smartphone, mais il existe de nombreuses alternatives, dont certaines sont plus anciennes.

Les principaux critères de choix sont :

- dimensions (adaptées à celles du smartphone pour optimiser le positionnement) ;
- lentilles en verre ou en résine (le verre est plus résistant) ;
- lentilles biconvexes ou plan-convexes (les lentilles biconvexes sont semble-t-il meilleures) ;

- focale (une focale plus courte donne un champ de vision plus large) ;
- diamètre des lentilles (des lentilles trop petites restreignent le champ de vision) ;
- système de fixation des lentilles (un support conique peut amener les lentilles plus près des yeux qu'un cadre plat) ;
- distance entre les lentilles ajustable à l'écart inter-pupillaire de l'utilisateur ;
- Sangle 3 points plutôt que 2 points (une sangle supplémentaire passant au dessus de la tête améliore le confort).

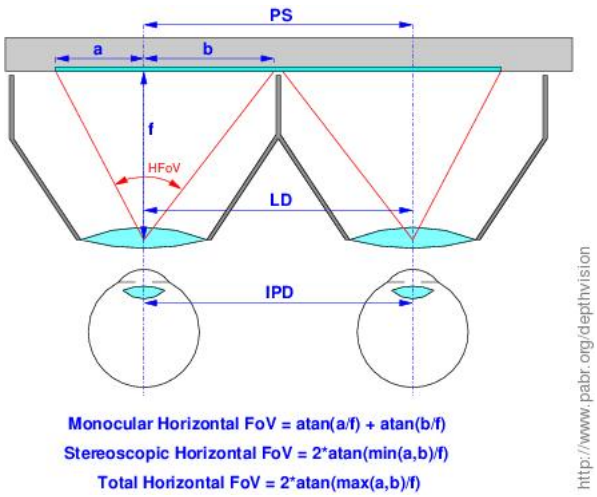
3. Remarques sur l'optique des casques stéréoscopiques

Il est facile d'infliger une première expérience de réalité virtuelle décevante à un utilisateur à cause d'un affichage stéréoscopique mal configuré. Aussi, il est choquant que la plupart des démonstrations en stéréoscopie *side-by-side* n'offrent pas la possibilité d'ajuster certains paramètres cruciaux.

Au minimum, il faut faire correspondre les distances suivantes :

- Écart inter-pupillaire (IPD) : Distance entre les centres des pupilles de l'utilisateur (autour de 63 mm). Cet écart est typiquement mesuré lorsque l'utilisateur regarde un objet lointain ; on peut adopter une valeur légèrement inférieure si la scène affichée est en champ proche.
- Écart inter-lentilles (LD) : Distance entre les centres des lentilles.
- Séparation des projections (PS) : Distance physique sur l'écran entre les centres optiques des projections 3D à gauche et à droite. C'est aussi approximativement la distance entre les rendus à gauche et à droite d'un objet lointain dans la scène 3D. Si le logiciel n'est pas en mesure de déterminer automatiquement les dimensions physiques des pixels, ceci nécessite une configuration manuelle.

  De nombreuses application se contentent de centrer les projections 3D au milieu de chaque demi écran. La séparation ne sera alors correcte que si la largeur de l'écran est égale à deux fois l'écart inter-pupillaire de l'utilisateur. Pour un écart de 63 mm cela correspond à un écran 16:9 de 5,7"). Avec un écran plus petit, la perspective sera incorrecte. Avec un écran plus grand, la séparation excessive provoquera une fatigue visuelle.



Ensuite, il est préférable que la longueur focale de la projection 3D corresponde à celle du casque, surtout lorsqu'on met en oeuvre du suivi de mouvements. Cela se résume à ajuster un facteur de mise à l'échelle 2D dans le logiciel.

Enfin, il est souhaitable de corriger la distorsion géométrique causée par les lentilles aux bords du champ de vision.

4. Logiciel

Le ASUS Xtion est supporté par OpenNI2 [http://structure.io/openni], que l'on peut compiler facilement pour Android.

Ce patch [depthvision-poc-1.patch] ajoute quelques lignes de C++ à `OpenNI2/Samples/SimpleRead/main.cpp` de sorte que le programme **SimpleRead** affiche des nuages de points en stéréoscopie directement dans `/dev/graphics/fb0`. Il s'agit seulement d'une démonstration de faisabilité sans prétentions. Ceci fonctionne sur le Galaxy S3 avec Android 4.3 rooté.

```
$ export NDK_ROOT=/path/to/android-ndk-r9
$ cd OpenNI2/Packaging
$ ./ReleaseVersion.py android
$ ls AndroidBuild/obj/local/armeabi-v7a/
$ adb push AndroidBuild/obj/local/armeabi-v7a/ ...
```

5. Application Android

Idéalement il faudrait convertir ce projet en une application Android ordinaire pour appareils non rootés. Des tests préliminaires suggèrent que c'est techniquement possible.

Bibliographie

[WXHMD] *WXHMD - Un casque vidéo sans fil sous Linux* . http://www.pabr.org/wxhmd/doc/wxhmd.fr.html .