

Capture, traitement et affichage d'images 3D

Objectifs

Ce laboratoire vise à explorer les fonctionnalités de la Kinect v2 et des écrans autostéréoscopiques Dimenco (2D+Depth). La SDK de la Kinect publiée par Microsoft permettra d'acquérir les données de la Kinect. Ces données seront traitées afin de répondre aux spécifications des écrans et ainsi être correctement affichées en 3D.

Contexte

La nouvelle Kinect v2 de Microsoft offre plusieurs fonctionnalités et améliorations par rapport au modèle précédent. Le système contient une caméra couleur 1080p, un émetteur infrarouge à précision accrue et un ensemble de microphones permettant de trouver la provenance du son. Depuis la version précédente, la technique employée par la Kinect pour percevoir la profondeur a changé, le système utilise maintenant le « *Time of flight* » versus la lumière structurée que le capteur utilisait auparavant ce qui permet d'avoir une précision de profondeur plus élevée.

Votre programme utilisera la caméra couleur ainsi que l'information de profondeur fournie par la Kinect pour afficher deux images contenant les informations nécessaires pour que les écrans Dimenco puissent afficher en 3D.

Quelques informations supplémentaires :

- Résolution de la caméra couleur : 1920x1080
- Résolution de la profondeur : 512x424
- Images par secondes : 30 (avec un bon éclairage, sinon 15)
- Résolution des écrans Dimenco : 3840x1080
- Portée de la Kinect : 0.5m à 4.5m

- Contraintes d'affichage des écrans Dimenco : (voir figure 1)
 1. Pour afficher du contenu 3D, vous devez afficher l'image couleur et la profondeur côte à côte.
 2. Ces images doivent être chacune à pleine résolution HD (1920x1080).
 3. L'image de profondeur doit être alignée avec l'image couleur (pas seulement les bordures, le contenu également).
 4. Dans le coin supérieur gauche de l'image couleur, vous devez insérer une image «entête».
 5. Ce contenu doit être plein écran dans votre affichage (pas de bordures).

Laboratoire

Vous devez concevoir un programme qui affiche les images provenant de la Kinect de façon à répondre aux exigences des écrans Dimenco pour afficher du contenu 3D. Votre programme devra rouler en continu (pas de traitement hors-ligne) et devra afficher une fréquence d'image raisonnable.

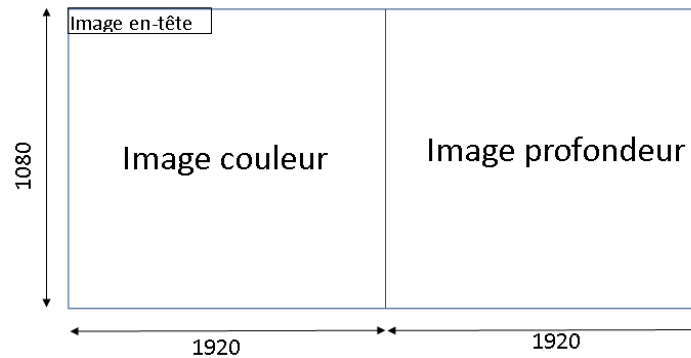


Figure 1 – Disposition des images, pour l'affichage 3D

Procédure générale

1. Connexion à la Kinect ;
2. Connexion aux senseurs de couleur et de profondeur ;
3. Acquisition d'une trame de chaque senseur ;
4. Traitement sur la trame de couleur ;
5. Traitement sur la trame de profondeur :
 - a. Conversion de l'information de profondeur en contenu visuel ;
 - b. Correction des différences physiques entre les deux caméras ;
 - c. Amélioration de la qualité de la profondeur ;
6. Affichage en format côte à côte des images ;
7. Insertion de l'image «entête» correcte dans le coin supérieur gauche ;
8. Affichage sur l'écran Dimenco ;
9. Répéter à partir de 3 ;

Environnement de travail

Pour ce laboratoire, un projet de base vous sera fourni. Ce projet est en C# sous l'environnement de développement Visual Studio sur Windows 10, et utilise la technologie XAML pour gérer l'interface d'affichage.

Ce projet de base effectue la connexion à la Kinect, ainsi que l'acquisition de l'image de couleur. Celle-ci est affichée deux fois à l'écran (en mode côte à côte). Ce programme affiche le résultat en plein écran et y insère une image « entête ». Cependant, le contenu de cette image «entête» n'est pas correct.

Afin de répondre aux objectifs du laboratoire, il faudra remplacer la seconde image couleur (celle de droite) par l'image de profondeur traitée et corriger le contenu de l'image « entête ».

Emgu cv (Wrapper d'OpenCV en C#) sera utilisé pour améliorer l'image de profondeur, et l'utilisation **Matlab** permettra la correction de l'image « entête ».

Les sections suivantes présentent les instructions nécessaires pour modifier et compléter correctement le projet de base fourni.

I. Connexion aux senseurs de couleur et de profondeur

Le programme de base permet la connexion à la trame de couleur.

Effectuer les modifications nécessaires afin de se connecter également au senseur de profondeur, tout en respectant la contrainte suivante : ne **pas** utiliser la classe `depthFrameSource`.

Indiquez dans le rapport la fonction utilisée

II. Acquisition et affichage de la couleur

Modifier l'acquisition et l'affichage de la trame de couleur, pour qu'ils soient cohérents avec les modifications apportées en section I.

III. Traitement de la profondeur

- a) Générer une trame de profondeur alignée

La résolution de la trame de profondeur et celle de la trame de couleur sont différentes. De plus, l'espace couleur et l'espace de profondeur sont aussi différents. Générez une trame de profondeur* alignée à celle de la couleur (alignement des bordures et contenu).

Indice : Utilisez une des méthodes de la classe **CoordinateMapper**

* Utiliser la classe **Image<,>** de Emgu CV pour la trame de profondeur, afin de faciliter le traitement par la suite (section V)

- b) Modifier le format de la valeur de profondeur

Le senseur de profondeur génère des valeurs de profondeur en millimètres, alors que l'écran autostéréoscopique s'attend à une carte de profondeur où les pixels les plus **proches** sont les plus lumineux (donc prêt de **255**) et les plus **loin** sont les moins lumineux (proche de **0**).

Proposez une formule de passage entre la profondeur en millimètres acquise par la Kinect et celle en niveau de gris compatible avec l'écran. Apportez cette modification à votre code et expliquez votre choix de formule dans votre rapport.

Indice : Vous pouvez vous inspirer des exemples de projet avec la Kinect (voir le SDK Browser), mais assurez-vous d'obtenir un résultat cohérent avec une carte de profondeur.

IV. Corriger le contenu de l'image «entête»

Au niveau du fichier `MainWindow.xaml`, une image «entête» nommée `EnTeteIncorrect.bmp` est insérée. Comme mentionné plus tôt, cette image ne contient pas le contenu correct pour que l'écran 3D reconnaisse le format 2D+Depth en format côte à côte.

Ci-dessous, les instructions pour générer l'image «entête» correcte :

Le message en tête (appelé *H*) contient 32 octets. Ce message doit être transformé en une image «entête». Celle-ci doit être insérée dans le coin haut gauche de l'écran.

La notation $H(x)^y$ désigne le y^{ieme} bit du x^{ieme} octet de H . ($x \in [0; 31]$ et $y \in [0; 7]$, 7 étant le bit de poids fort).

L'image en tête (voir figure 2) contient l'information du message H au niveau des sous-pixels **bleus** (appelés B).

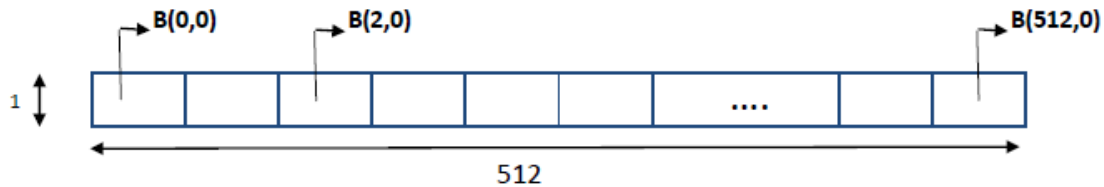


Figure 2- Taille et indexation de l'image en tête

Le passage message/image se fait selon la formule suivante :

$$B(2 * (7 - y) + 16 * x, 0)^7 = H(x)^y$$

Dans notre cas (Image 2D+Depth, côte à côte), le contenu de H est (en Hexadécimal) :

[illegible]

Nommez l'image en tête générée `EnTeteModifiee.bmp` et modifiez le fichier `MainWindow.xaml` pour qu'il affiche la nouvelle image.

Il est conseillé d'utiliser **Matlab** pour effectuer le passage message/image, mais vous pourrez utiliser un autre outil si vous voyez plus de facilité (cependant, vous pourriez avoir moins de support de la part du chargé de laboratoire).

V. Améliorer la profondeur obtenue

Si toutes les sections précédentes ont été correctement effectuées, l'effet 3D devrait s'afficher sur l'écran autostéréoscopique.

On devrait aussi remarquer que l'image de profondeur obtenue présente plusieurs défauts. En effet, celle-ci contient beaucoup de bruit ainsi que plusieurs zones noires (trous), elle est aussi assez instable (c'est-à-dire deux trames qui se suivent, sont très différentes).

Le but de cette section est de traiter l'image de profondeur afin d'obtenir un meilleur rendu 3D. Il existe plusieurs façons de procéder pour améliorer l'image de profondeur. Vous devrez **proposer** un algorithme d'amélioration de profondeur, pour ce faire explorez les traitements que permet Emgu CV.

Expliquez et justifiez dans votre rapport l'algorithme d'amélioration utilisé.

Références utiles

- Vidéos d'introduction à la Kinect v2, ils ont été publiés au lancement de la nouvelle Kinect et démontrent comment utiliser les différentes capacités de la Kinect. Ils présentent des exemples de code (qui sont parfois invalide, car le SDK a été modifié depuis). Vidéos suggérés en ordre d'importance :
 - Programming Kinect for Windows v2: (02) Kinect Data Sources and Programming Model
 - Programming Kinect for Windows v2: (01) Introducing Kinect Development<http://www.microsoft.com/en-us/kinectforwindows/develop/how-to-videos.aspx>
- Documentation MSDN du SDK pour la Kinect
<https://msdn.microsoft.com/en-us/library/dn758675.aspx>
- Exemples de projets C#/C++ fonctionnant avec la Kinect
 - Color, Depth, CoordinateMapping(Voir le SDK Browser)
- Votre chargé de laboratoire
- Documentation de la librairie Emgu CV
<http://www.emgu.com/wiki/files/3.1.0/document/html/8dee1f02-8c8a-4e37-87f4-05e10c39f27d.htm>

Date de remise

La remise complète devra être effectuée sur MOODLE avant le **11 juin 2017 à 23 h 55**.

Cependant compte tenu du nombre limité d'écrans disponibles, la quatrième période de laboratoire sera dédiée aux démonstrations de chaque équipe, alors assurez-vous de pouvoir afficher vos images avant le début de la quatrième période.

Livrables

- La solution complète Visual Studio (tout le dossier contenant le fichier .sln) avec les références établies.
 - Votre solution doit pouvoir être exécutée directement sans étape supplémentaire.
 - Faites attention s'il y a des chemins de fichiers absolus dans votre code !
- Le programme Matlab (ou autre, si applicable) qui permet de générer l'image «entête».
- Un rapport sous format PDF
 - Votre rapport devra détailler les traitements effectués à chaque section (de I à V), et justifier les choix effectués
 - Votre rapport devra expliquer les particularités que vous avez rencontrées en développant avec la Kinect (ex. Limitations de la Kinect, du SDK, des écrans Dimenco) et discuter des résultats obtenus.
 - Votre rapport devra aussi inclure une introduction, une conclusion et un diagramme «Flowchart» représentant votre programme

- Pour le faire, basez-vous sur la « Procédure générale » plus haut, et les traitements effectués dans chacune des cinq sections.
- Une démonstration en classe, au plus tard durant la quatrième période de laboratoire.
 - Vous devrez prouver que votre application fonctionne.

Barème de correction

- Code (40%)
 - I. Connexion aux senseurs de couleur et de profondeur **3%**
 - II. Acquisition et affichage de la couleur **2%**
 - III. Traitement de la profondeur
 - a) Générer une trame de profondeur alignée **10 %**
 - b) Modifier le format de la valeur de profondeur **5%**
 - IV. Corriger le contenu de l'image «entête» (avec programme Matlab) **10 %**
 - V. Améliorer la profondeur obtenue **10%**
- Rapport (35%)
 - Détails et justifications des traitements de chaque section :
Section I **1%** - Section II **1%** - Section III a) **5%** - Section III b) **5%** - Section IV **3%** - Section V **10%**
 - Particularités rencontrées et discussion des résultats **5%**
 - Introduction, conclusion, diagramme *Flowchart*, bonnes organisation du rapport et clarté de la langue **5%**
- Preuve du fonctionnement de l'application (25%)
 - Activation de la 3D Dimenco **10%**
 - Qualité de la 3D affichée **10%**
 - Fréquence d'image raisonnable **5%**

Procédure de remise

Remise sur MOODLE