



**École de technologie supérieure**  
*Département de génie logiciel et des TI*

## **Capture, traitement et affichage d'images 3D**

|                              |                               |
|------------------------------|-------------------------------|
| <b>N° du laboratoire</b>     | 1                             |
| <b>N° d'équipe</b>           | 09                            |
| <b>Étudiants</b>             | André Tah<br>Alexandre Forget |
| <b>Cours</b>                 | GTI780                        |
| <b>Session</b>               | Été 2018                      |
| <b>Groupe</b>                | 01                            |
| <b>Chargé de cours</b>       | Carlos Vázquez                |
| <b>Chargé de laboratoire</b> | Roseline Olory Agomma         |
| <b>Date</b>                  | 07 Juin 2018                  |

# **Table des matières**

**Introduction**

**Connexion aux senseurs de couleur et de profondeur**

**Acquisition et affichage de la couleur**

**Traitement de la profondeur**

**Corriger le contenu de l'image « entête »**

**Améliorer la profondeur obtenue**

**Conclusion**

**Références**

# Introduction

Le cours GTI780 s'articule autour de l'étude des aspects nouveaux et d'intérêt majeur en technologies de l'information et est reparti en deux parties, entre autre les systèmes de vision 3D et la réalité virtuelle. Ce laboratoire s'inscrit dans les Systèmes de vision 3D et a pour but de permettre aux étudiants d'explorer les fonctionnalités de la Kinect v2 et des écrans auto stéréoscopiques Dimenco (2D+Depth). Il s'agira de capturer une vidéo en couleur via la kinect v2 et d'exposer cette vidéo vis-à-vis de la même vidéo mais cette fois-ci en noir et blanc. L'écran Dimenco se chargera d'afficher les données en 3D lorsque le bon entête sera implémenté dans le code de traitement. Pour réaliser tout cet ensemble, nous suivrons les différentes étapes énumérées dans le descriptif du laboratoire à savoir :

- Connexion aux senseurs de couleur et de profondeur
- Acquisition et affichage de la couleur
- Traitement de la profondeur
- Corriger le contenu de l'image « entête »
- Améliorer la profondeur obtenue

## **I. Connexion aux senseurs de couleur et de profondeur**

Pour respecter la contrainte de ne pas utiliser la classe DepthFrameSource, il restait une option évidente, utiliser un MultiFrameSource. Le code a été modifié pour passer par une MultiFrameSource reader pour ensuite traiter respectivement les images de couleurs et de profondeur

## **II. Acquisition et affichage de la couleur**

Une grande partie du code présent dans Reader\_FrameArrived a été transmis à ColorSection qui est responsable de créer la section avec l'image de couleurs

## **III. Traitement de la profondeur**

La méthode d'entrée pour le traitement de la profondeur est la méthode DepthSection, celle-ci est responsable de générer les DeathSpacePoint depuis l'image de couleur, de transformer le depthFrame en Bitmap et d'appliquer le bitmap à la section des profondeurs.

### a) Générer une trame de profondeur alignée

```
void DepthSection(DepthFrame frame)
{
    using (KinectBuffer depthFrameData = frame.LockImageBuffer())
    {
        this.coordinateMapper.MapColorFrameToDepthSpaceUsingIntPtr(
            depthFrameData.UnderlyingBuffer,
            depthFrameData.Size,
            this.colorMappedToDepthPoints);
        PictureBox2.Source = getImageSourceFromBitmap(DepthFrameToBitmap(frame));
    }
}
```

La dernière image représente le code effectué pour utiliser CoordinateMapper pour retrouver les DepthSpacePoints. Même si le résultat final utilisé est d'étirer l'image, voici ci-dessus la méthode CoordinateMapper.

### b) Modifier le format de la valeur de profondeur

Pour modifier le format de la valeur de profondeur de millimètres vers (0 à 255), il fallait trouver la valeur d'intensité à appliquer à la couleur RGB. Inévitablement que R, G et B ont la même intensité. Pour trouver la valeur d'intensité, il est important d'évaluer la valeur la plus petite de profondeur soit minDepth et la valeur la plus haute soit maxDepth. Un élément à mentionner est la valeur intitulée mapDepthToByte qui équivaut à  $1000/256$  et qui aide pour le ratio millimètre/(byte ou 0 à 255). L'intensité est déterminée selon le ratio de profondeur avec mapDepthToByte.



# Conclusion

Dans ce premier laboratoire concernant les systèmes de vision 3D, il était surtout question d'explorer les fonctionnalités de la Kinect v2 et des écrans auto stéréoscopiques Dimenco. De façon chronologique, nous avons effectué des modifications nécessaires pour nous connecter au senseur de profondeur en vue de traiter la profondeur de l'image. Ensuite, nous avons corrigé l'image « entête » afin de permettre à l'écran Dimenco d'afficher en 3D. Enfin, nous avons essayé d'améliorer la profondeur obtenue lors des différentes modifications apportées mais le temps imparti pour ce laboratoire nous a semblé un peu court pour achever cette tâche. Par ailleurs, ce laboratoire nous a permis de comprendre les fonctionnalités de Kinect v2.

# Référence

<https://www.kevinmarburger.fr/wp-content/uploads/2015/06/Rapport-de-fonctionnement-Kinect-2.pdf>

<https://www.mathworks.com/help/daq/ref/hextobinaryvector.html#btk0k37-1-hexNumber>

<https://www.mathworks.com/help/images/ref/im2bw.html>

<http://www.matrixlab-examples.com/hex-to-binary.html>

<https://www.mathworks.com/matlabcentral/fileexchange/34713-image-to-hex--hex-to-image-conversion>

<https://www.mathworks.com/help/matlab/ref/image.html>

<https://www.mathworks.com/matlabcentral/answers/224037-how-to-convert-a-binary-code-to-image>