Département d'informatique

Faculté des sciences

Université de Sherbrooke

Rapport d'avancement de projet

par

Benjamin Provost – prob2702

Étienne Lacasse - lace3304

Remis à

Djemel Ziou et Vincent Ducharme

Dans le cadre de l'activité pédagogique

Projet d'informatique - IFT592

Sherbrooke

26 octobre 2021

Le projet

Le projet auquel nous nous sommes attaqués est un projet de reconstruction de buste digital en 3 dimensions à l'aide de la technologie de la Kinect première génération pour la Xbox 360. L'objectif du projet est d'améliorer la précision de l'acquisition des données par la Kinect afin d'obtenir le modèle le plus fidèle et le plus constant possible.

Chronologie

Démarrage cahoteux

À l'issue de la mi-session, nous contemplons ce qui a été réalisé depuis le début du projet pour réaliser que nous avons uniquement été capables d'arriver à quelque chose de fonctionnel après sept semaines de travail.

Durant les premières semaines, la majorité de notre temps a été investi sur la planification du projet ainsi que la lecture de documentation afin de mieux comprendre la technologie sur laquelle repose la Kinect.

Après trois semaines, nous étions prêts à nous lancer. Malheureusement, le destin en a voulu autrement. Premièrement, l'ordinateur qui devait nous être fourni par le département d'informatique a pris plus de temps à arriver, nous ne l'avons toujours pas. De plus, il nous était impossible d'installer quelque programme que ce soit sur les ordinateurs du laboratoire, car nous n'avions pas les droits d'administrateurs sur ces machines. Ainsi, nous avons été dans l'obligation d'utiliser nos ordinateurs portables personnels pour commencer. Le problème de cette solution est qu'un d'entre nous possède un MacBook et l'autre un PC Windows. Les pilotes officiels pour la Kinect sont produits par Microsoft pour Windows seulement. Nous n'avions donc qu'un seul ordinateur sur lequel nous pouvions faire du développement.

Une lancée prometteuse

Suite à nos recherches par rapport à la compatibilité de la Kinect de première génération avec Windows, nous avions installé le « Kinect for Windows Software Development Kit » version 1.8 ainsi que le « Kinect for Windows Developer Toolkit 1.8 ». Suite à cette installation, nous avions accès à Kinect Studio qui nous offrait plusieurs échantillons de projets qui nous ont permis d'en découvrir plus sur les possibilités que nous offrait la Kinect telles la reconnaissance faciale, la reconnaissance de squelette et le retrait d'arrière-plan. Suite à la connexion réussie avec la Kinect, nous avons trouvé plusieurs programmes de scan qui servent à atteindre le même but que ce que nous voulons réaliser. Nous avons testé des scanners tels que Shapify.me, Skanect et ReconstructMe. Malheureusement, seul ReconstructMe nous avait donné un résultat concluant, soit un modèle complet avec couleurs qui avait du sens, car tous les autres scanners avaient une faiblesse quelconque. Pour Shapify.me, celui-ci devait se connecter à un serveur distant pour la génération de modèle, cependant le port était bloqué par les serveurs de l'université. Pour Skanect, la génération était facile et rapide, mais nous ne pouvions exporter que 500 faces avec la version gratuite du logiciel. En conclusion, il a été vraiment intéressant d'explorer les différentes versions de scanners disponibles sur le marché. Cela nous servira de comparatif avec notre propre génération de modèle pour voir les lacunes de notre programme par rapport à ce qui est commercialement disponible.

Casse-tête

Pour remédier à nos problèmes de système d'exploitation, nous avons tenté d'utiliser des pilotes libres tels que OpenKinect. Cette solution fonctionnait bien avec macOS, mais, malencontreusement, c'était hasardeux sur Windows, car plusieurs librairies devaient être installées manuellement. Ces installations étaient souvent problématiques, car elles dépendaient de ressources qui n'avaient pas été mises à jour depuis plusieurs années et n'étaient donc tout simplement pas installables.

Après une petite rencontre avec Pr Ziou et un de ses étudiants ayant travaillé avec cette technologie auparavant, nous avons décidé de rester avec ce qui était offert par Microsoft et donc de programmer en C#. Par contre, ce qui nous avait été suggéré était d'utiliser la version 2.0 de « Kinect for Windows Software Development Kit » qui s'est révélée incompatible avec notre périphérique, car celle-ci avait été développée uniquement pour la deuxième génération de Kinect. Après un peu de bricolage pour désinstaller tous ces programmes désormais inutiles, nous sommes revenus à la version 1.8 du logiciel, ce qui nous a permis de commencer à développer.

Début du code

Après avoir décidé du pilote et du langage que nous allions utiliser, nous avons effectué un peu plus de recherche et nous avons découvert la librairie « Kinect Fusion » qui permet de créer un modèle 3D à l'aide des données générées par notre Kinect. Ainsi, en partant de cette base, nous avons fait un premier programme permettant de générer un objet 3D de ce que la Kinect voit.

Ce que le futur nous réserve

Maintenant que nous avons un projet qui nous permet de générer un modèle en 3 dimensions à l'aide de la Kinect, plusieurs tests et modifications devront être faits pour améliorer la qualité du résultat produit. Par exemple, il nous faut ajouter la possibilité d'exporter le modèle produit afin de pouvoir le conserver et l'afficher dans un espace virtuel quelconque. De plus, il serait intéressant que notre programme coupe ce qui se trouve en dessous du buste de la personne scannée, car au final, c'est tout ce que nous voulons. Ensuite, ajouter un mode de génération de modèle à partir de multiples prises de vue statiques plutôt que d'utiliser une génération continue permettrait de simplifier grandement l'utilisation et donnerait des résultats plus constants. Pour arriver à faire une génération à partir de prises de vues, il sera nécessaire de tester quel est l'angle maximum nécessaire afin de ne pas perdre trop de qualité sur le modèle produit. La même chose devra être faite pour la capture en continu, car il nous faut trouver la

vitesse maximale à laquelle un utilisateur peut tourner devant la caméra sans que la qualité du résultat en soit impactée. Une autre amélioration qui pourrait être faite est l'utilisation de EmguCV, la version C# de OpenCV, afin de retirer de la matrice de profondeurs les valeurs qui font partie de l'arrière-plan pour améliorer la qualité du résultat afin que celles-ci ne soient pas prises en compte lors de la génération du modèle et y soient incluses par erreur. Nous allons aussi devoir tester notre programme avec diverses variables, tels les matériaux de vêtements, les accessoires que pourraient porter les utilisateurs ainsi que les divers types de caractéristiques physiques de l'utilisateur selon leur ethnicité et leur sexe afin d'évaluer son comportement de la librairie de reconstruction et de la technologie infrarouge d'obtention de la profondeur.