

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet Recherche & Développement 2018-2019

Deep learning et réalité augmentée

**POLYTECH[®]**
TOURS

Entreprise
Polytech Tours



Tuteur entreprise
GILLES VENTURINI (Professeur d'Informatique)

Étudiant
BOLUN ZHANG (DI5)

Tuteur académique
GILLES VENTURINI

Liste des intervenants

Entreprise

Polytech Tours
64 avenue Jean Portalis, 37200 Tours, FRANCE
www.univ-tours.fr/



Nom	Email	Qualité
BOLUN ZHANG	bolun.zhang@etu.univ-tours.fr	Étudiant DI5
GILLES VENTURINI	gilles.venturini@etu.univ-tours.fr	Tuteur académique, Département Informatique
GILLES VENTURINI	gilles.venturini@etu.univ-tours.fr	Tuteur entreprise, Professeur d'Informatique



Avertissement

Ce document a été rédigé par BOLUN ZHANG susnommé l'auteur.

L'entreprise Polytech Tours est représentée par GILLES VENTURINI susnommé le tuteur entreprise.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par GILLES VENTURINI susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

BOLUN ZHANG, *Deep learning et réalité augmentée :* , Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2018-2019.

```
@mastersthesis{
  author={ZHANG, BOLUN},
  title={Deep learning et réalité augmentée : },
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2018-2019}
}
```

Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iv
1 Introduction	1
1 Introduction.....	1
2 Contexte de la réalisation.....	1
3 Objectifs.....	2
4 Hypothèses	2
5 Bases méthodologiques	2
2 Description générale	3
1 Environnement du projet	3
2 Caractéristiques des utilisateurs	3
3 Fonctionnalités du système	3
4 Structure générale du système	5
4.1 Programme version 1 - Reconnaître les images augmentées.....	5
4.2 Programme version 2 - Reconnaître l'objet 3D	5
4.3 Programme version 3 - Reconnaître plusieurs objets 3D et ajouter des interactions	6

3	Etat de l'art	7
1	La bibliothèque ARCore.....	7
1.1	Introduction.....	7
1.2	Comment ARCore fonctionne.....	8
1.2.1	Entrée de flux vidéo.....	8
1.2.2	Reconnaître les images augmentée / surfaces.....	8
1.2.3	Faire correspondre les objets reconnu et les fichiers de ressources	9
1.2.4	Compréhension de l'environnement.....	10
1.2.5	Estimation de la lumière	10
1.2.6	Ancre et traçables.....	10
1.2.7	Sortie de flux vidéo.....	11
1.3	Interaction utilisateur.....	11
2	La bibliothèque ARToolKit.....	12
3	Pourquoi choisir ARCore.....	13
4	Analyse et conception	15
1	Image augmentée et base de données d'images augmentée.....	15
1.1	Image augmentée.....	15
1.2	L'outil acoreimg	15
1.2.1	Vérifier la qualité de l'image.....	16
1.2.2	Créer un fichier de base de données d'images.....	16
1.3	Limite de Reconnaissance.....	16
1.4	Base de données d'images augmentée.....	17
2	Modélisation logicielle	17
2.1	Modèle.....	17
2.1.1	Reconnaître les images augmentée	17
2.1.2	Reconnaître d'objet 3D	18
2.1.3	Reconnaître plusieurs objets 3D et ajouter des interactions	19
5	Mise en oeuvre	21
1	Mise en oeuvre du semestre 9	21
1.1	Utiliser l'outil arcoring	21
1.1.1	Prendre les images augmentée	21
1.1.2	faire l'évaluation de les images augmentée.....	22
1.1.3	Faire la création et l'utilisation de bases de données d'images augmentée.....	23
1.2	Utiliser l'outil Sceneform.....	24
1.2.1	Importer et créer des modèles 3D	24
2	Mise en oeuvre du semestre 10	26
2.1	Reconnaître les images augmentées	26

2.2	Reconnaître l'objet 3D	27
2.3	Reconnaître plusieurs objets 3D et ajouter des interactions.....	28
2.3.1	Ajouter plus d'objets 3D reconnaissables	28
2.3.2	Ajouter plus d'interactions.....	29
2.3.3	Ajouter une interface de programme	30
6	Bilan et conclusion	33
1	Bilan du semestre 9	33
2	Bilan du semestre 10	33
3	Reste à faire.....	34
4	Test.....	34
5	Conclusion	35
	Annexes	36
A	Planification	37
1	Aperçu de gestion de projet	37
2	Aperçu de gestion de projet	37
B	Technologies utilisées	40
1	Outils pour la gestion de projet	40
2	Technologies pour le développement	40
C	Spécifications fonctionnelles	42
D	Spécifications nonfonctionnelles	43
1	Contraintes de développement et conception	43
2	Contraintes de fonctionnement et d'exploitation	43
2.1	Performances.....	43
2.2	Capacités.....	43
2.3	Modes de fonctionnement	44

Table des figures

2 Description générale

1	Diagramme de cas d'utilisation	4
2	Schéma du SI	5
3	Schéma du SI version 2.....	6

3 Etat de l'art

1	Logo ARCore.....	7
2	Processus de reconnaissance d'ARcore	8
3	Le suivi d'ARCore.....	9
4	L'effet d'ARCore.....	9
5	Compréhension de l'environnement.....	10
6	Les résultats obtenus par ARCore.....	11
7	Logo ARToolKit	12
8	Processus de reconnaissance d'ARToolKit	12
9	L'effet d'ARToolKit	13
10	L'effet de détection de lumière d'ARCore	14

4 Analyse et conception

1	La commande pour évaluer l'image.....	16
2	Norme de notation pour les images augmentées.....	16
3	La commande pour créer la base de données d'images.....	16
4	La structure d'application version 1.0	18
5	La façon de prendre les images améliorée	19
6	La structure d'application version 3.0	19

5 Mise en oeuvre

1	La façon de capturer une image augmentée	22
2	Différents angles pour image augmentée	22
3	Les images augmentées	22
4	Bat script utilisant l'outil arcoring	23
5	Résultats d'évaluation pour les images	23
6	La méthode AugmentedImageDatabase()	24
7	Le plug-in Google Sceneform Tools (Beta)	24
8	Import Sceneform Asset	25
9	Afficher le ressource du modèle 3D	25
10	La méthode SetupAugmentedImageDb()	26
11	La méthode augmentedImageDatabase.addImage()	26
12	La méthode loadAugmentedImageBitmap()	26
13	La méthode onUpdateFrame()	27
14	Le programme Version 1.0	27
15	Le programme Version 2.0	28
16	Modifier la méthode SetupAugmentedImageDb()	29
17	Modifier la méthode onUpdateFrame()	29
18	La méthode videoPlayer()	30
19	Le dossier <raw>	30
20	Le menu principal	30
21	L'interface pour présenter les objets reconnaissable	31
22	Effet de reconnaissance finale du programme 1	31
23	Effet de reconnaissance finale du programme 2	32
24	Effet de reconnaissance finale du programme 3	32

6 Bilan et conclusion

1	Le menu <Record Espresso Test>	34
2	Le fichier <WelcomeActivityTest>	34
3	La fenêtre <Record Your Test>	35

A Planification

1	Le diagramme de Gantt	37
2	Les tâches dans le diagramme de Gantt	38

B Technologies utilisées

1	Gitkraken	40
---	-----------------	----

1

Introduction

Dans ce chapitre, on décrit le contexte du projet et le structure du rapport ainsi que les acteurs, les objectifs, l'hypothèses, et les bases méthodologiques sur la gestion de projet.

1 Introduction

Le projet de recherche et développement, comme le projet de fin d'étude, est le dernier cycle dans la vie d'étudiant, et c'est très important et obligatoire pour que les étudiants obtiennent leurs diplômes d'ingénieur. Il peut intégrer toutes les connaissances et les techniques que les étudiants apprennent de leurs formations, et mettre tous leurs acquis à la pratique.

Ce projet est encadré par M.Gilles Venturini, enseignant chercheur à Polytech Tours, et est exécuté par Bolun ZHANG, élèves ingénieurs en 5ème année de spécialité Informatique à Polytech Tours. Le client du projet est M.Gilles Venturini.

Dans ce rapport, le chapitre 1 décrit le contexte de la réalisation, y compris l'objectif du logiciel, l'hypothèse et la méthodologie utilisée. Le chapitre 2 présente les descriptions générales du logiciel sur l'environnement, les utilisateurs, le structure, l'interface, et on expose en détail la fonctionnalité et la non-fonctionnalité. Le chapitre 3 présente les techniques existantes relatives à ce projet. Le chapitre 4 explique les analyses de problème et montre la modélisation mathématique et la modélisation logicielle. Le chapitre 5 explique en détail l'algorithme proposé et des améliorations. Les bilans pour ce projet dans le chapitre 6.

2 Contexte de la réalisation

Le projet vise à fournir un logiciel (en option utilisant un réseau de neurones profonds) pour implémenter la réalité augmentée. Il est conçu pour permettre aux clients de reconnaître des objets 3D et de fournir des interactions de réalité augmentée via l'application mobile (plateforme Android). L'utilisateur peut observer une image en réalité augmentée de l'objet 3D dans l'espace réel de l'application mobile et, en cliquant sur l'élément, peut fournir des informations visuelles (type, informations détaillées de l'objet...).

3 Objectifs

Ce projet consiste à étudier, implémenter et tester un système permettant de superposer des informations sur des scènes réelles. Le système utilise une stratégie en deux temps :

1) à partir d'un ensemble d'images d'objets (par exemple des œuvres d'arts), il s'agit de reconnaître dans un premier temps quelle est l'œuvre observée par le téléphone/tablette de l'utilisateur. À cet égard, nous utiliserons les points caractéristiques pour identifier les objets de la même manière que les objets 2D, tandis que pour les objets 3D, nous utiliserons plus de points caractéristiques pour permettre la reconnaissance sous différents angles, positions, intensité lumineuse et distance.

2) ensuite, une fois l'œuvre bien identifiée, on charge des images/textures de l'œuvre dans un composant de réalité augmentée. Ces textures sont alors utilisées par la librairie d'AR afin de délivrer un contenu numérique (image, sons, animations 3D, etc). On pourra utiliser une bibliothèque de réalité augmentée comme ARCore.

4 Hypothèses

L'utilisateur utilise l'application pour reconnaître des objets 3D réels et utilise des fonctions de réalité augmentée pour comprendre et interagir avec des objets 3D. L'application est basée sur une bibliothèque de réalité augmentée ARCore. Reconnaître par la caméra du téléphone/tablette Android.

Nous pensons qu'un objet 3D peut être résumé dans une collection de plusieurs images augmentées, c'est-à-dire qu'un ensemble d'images augmentées prises sur le même 3D objet dans différentes directions, différentes distances, différentes intensités lumineuses, etc., peut déterminer un objet 3D unique situé dans un espace spécifique.

En utilisant un ensemble d'images augmentées d'un objet 3D spécifique, nous pouvons utiliser ARCore pour suivre chaque image augmentée en temps réel afin de suivre et d'identifier des objets 3D spécifiques.

5 Bases méthodologiques

- Pour la gestion de projet, on utilise le diagramme de Gantt pour contrôler le déroulement du projet.
- Pour la modélisation logicielle, on utilise les diagrammes d'UML.
- Pour la modélisation mathématique et la rédaction de rapport, on utilise « LaTeX ».
- Pour l'IDE on utilise Android Studio.

Pour le semestre 9, ce que on doit faire est la recherche sur notre projet. Cela correspond aux phases de l'étude de faisabilité à l'analyse. Pour le semestre 10, nous devons suivre la conception du semestre 9 et atteindre les objectifs attendus.

2

Description générale

1 Environnement du projet

Le projet fonctionne comme suit : un ou plusieurs clients utilisent leur appareil mobile (téléphone mobile, tablette Android) pour utiliser l'application mobile. Cette application reconnaît les objets 3D via Arcore et génère un modèle de réalité augmentée prédéterminé. L'utilisateur peut identifier tous les objets 3D stockés dans la base de données d'images via l'application et interagir avec la réalité augmentée et les objets 3D.

L'environnement de développement consiste aux points suivants :

- Le système d'exploitation : Windows ;
- Le langage de programmation : JAVA avec Android development Kit ;
- L'IDE pour le développement : Android Studio ;

2 Caractéristiques des utilisateurs

Ce logiciel a un seul type d'utilisateur.

Client : Ce sont les principaux utilisateurs de l'application. Ils utilisent l'application dans des lieux spécifiques (musées, galeries, etc.) et réalisent des interactions basées sur la réalité augmentée avec des éléments spécifiques (sculptures, expositions, etc., objets 3D). L'application doit être facile à utiliser et les clients peuvent l'utiliser sans connaître la technologie ou la formation. L'application est disponible pour tous les niveaux, tous les âges.

3 Fonctionnalités du système

Client :

- Connection à la application :
L'utilisateur ouvre l'application mobile et entre dans l'interface principale. L'utilisateur peut sélectionner une autre interface à utiliser.

- Afficher la base de données d’images augmentée :
Grâce à cette interface, l’utilisateur peut visualiser les objets 3D reconnaissables qui ont été enregistrés dans la base de données d’images augmentée, ce qui permet de connaître les objets 3D que le programme peut actuellement reconnaître et avec lesquels il peut interagir.
Dans cette interface, tous les objets 3D reconnaissables seront représentés par l’une de ses images augmentées et seront accompagnés d’une description textuelle correspondante, ce qui aidera l’utilisateur à visualiser et à comprendre l’objet et à lui permettre de savoir quels objets doivent être utilisés sur ce programme pour effectuer des interactions en réalité augmentée.
- Suivre des objets 3D(L’objectif principal) :
Avec ARcore, les utilisateurs peuvent reconnaître et interagir avec des objets 3D et utiliser l’objet que ils souhaitent connaître avec le support de la Réalité Augmentée pour afficher des informations sur celui-ci.
Grâce à la caméra de l’appareil mobile, l’application suit en permanence l’objet 3D reconnaissable situé dans l’espace réel et superpose l’objet de réalité augmentée correspondant sur l’objet 3D reconnu. L’utilisateur peut ainsi observer l’objets Réalité Augmentée (ex.vidéo, intro, animation, modèles 3D, etc.) à l’aide de l’application mobile et obtenir des informations sur l’objet.
- Ajouter de nouvelles images augmentée :
Pour l’application, les utilisateurs peuvent ajouter de nouvelles images augmentées à la base de données d’images augmentées afin que le programme puisse reconnaître davantage d’objets 3D. L’utilisateur peut faire des suggestions pertinentes aussi, telles que le fait de demander des erreurs ou des problèmes, et le programme comportera une zone et une fonction permettant à l’utilisateur de suggérer des améliorations.

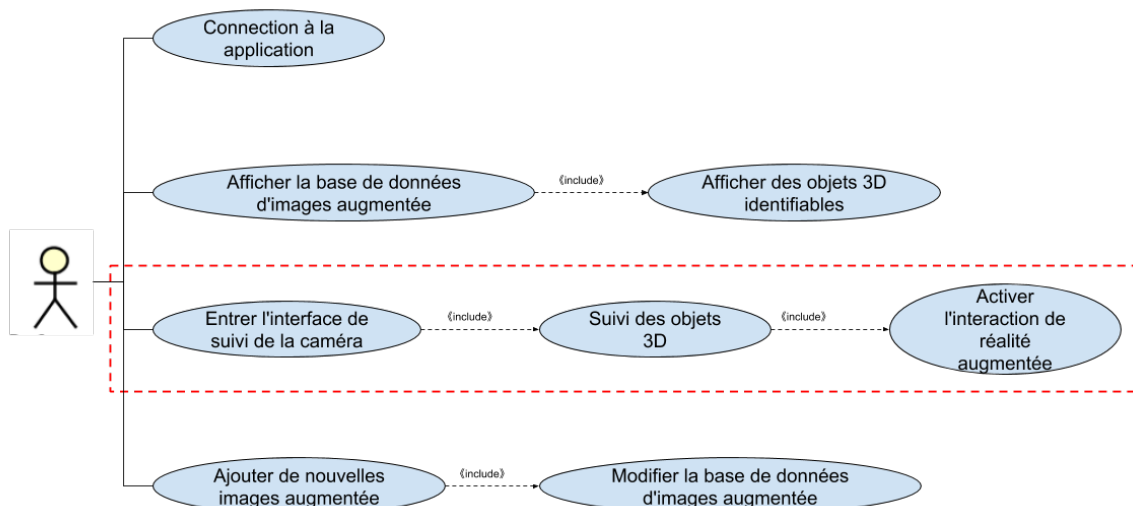


Figure 1 – Diagramme de cas d'utilisation

4 Structure générale du système

4.1 Programme version 1 - Reconnaître les images augmentées

Pour le premier objectif, nous devons utiliser ARCore pour reconnaître l'image augmentée. La première version du programme ne nécessite pas de vérification de l'identification : les objets pouvant être identifiés sont préalablement enregistrés dans la base de données d'images fournie avec le programme. Par conséquent, le système est divisé en trois parties :

- Application mobile : fournit une interface utilisateur permettant aux utilisateurs d'interagir avec la réalité augmentée d'objets 3D à l'aide d'une application mobile.
- ARCORE : l'application mobile implémente la reconnaissance d'objet et la possibilité de générer des objets interactifs en appelant ARCORE et renvoie les résultats à l'application mobile.
- Base de données d'images (BDI) : la base de données d'images stocke l'image de la fonction de l'objet à reconnaître par ARcore ARCORE doit acquérir l'image de l'objet 3D et rechercher l'image de la fonction correspondante dans la base de données d'images pour identifier l'objet 3D.

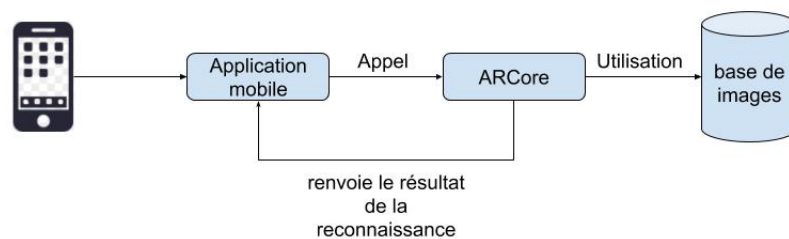


Figure 2 – Schéma du SI

4.2 Programme version 2 - Reconnaître l'objet 3D

La deuxième version du programme reconnaîtra l'objet 3D et la base de données d'images augmentée contiendra les images augmentée pris par différentes directions, des angles différents et de différentes distances de ce objet 3D, qui seront utilisées ensemble pour identifier le même objet 3D.

Une fois l'objet 3D identifié, le programme implémentera également une interaction principale (par exemple, placer un modèle 3D sur l'objet).

La figure de deuxième version du programme qui reconnaît les objets 3D et permet aux utilisateurs d'interagir avec eux. En reconnaissance l'image augmentée stockée dans la base de données d'images augmentée et en contactant les ressources dans la base de données de ressources, le programme renverra les résultats au dispositif mobile, mettant ainsi en œuvre le processus de reconnaissance-> interaction.

Le schéma de structure de programme de la deuxième version du programme est le suivant :

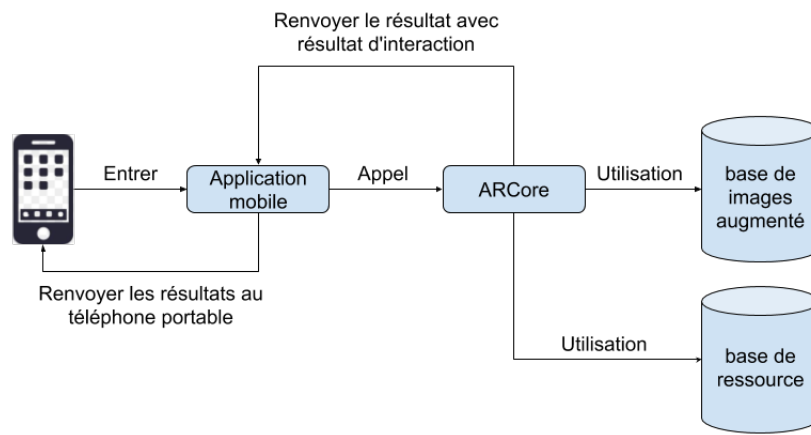


Figure 3 – Schéma du SI version 2

4.3

Programme version 3 - Reconnaître plusieurs objets 3D et ajouter des interactions

La troisième version du programme est fondamentalement la même que la deuxième version, mais la troisième version pourra suivre et interagir avec plusieurs objets 3D en même temps. Le programme utilisera donc les images augmentée de la base de données plus fréquemment et constamment. Les ressources de la base de données de ressources sont liées pour permettre l'identification et l'interaction de plusieurs objets 3D.

3

Etat de l'art

Dans ce chapitre, nous décrivons les développements actuels de certains outils et techniques utilisés dans le projet.

1 La bibliothèque ARCore

1.1 Introduction



Figure 1 – Logo ARCore

ARCore est la plateforme de construction d'expérience de réalité augmentée de Google. ARCore utilise différentes API pour informer votre téléphone de son environnement, comprendre le monde réel et interagir avec les informations. Certaines API disponibles sur Android et iOS prennent en charge l'expérience AR partagée.

ARCore utilise trois fonctionnalités principales pour intégrer du contenu virtuel au monde réel vu à travers une caméra mobile :

- Suivi des mouvements : laissez le téléphone comprendre et suivre sa position par rapport au monde réel.
- Compréhension de l'environnement : permet au téléphone de détecter la taille et la position de diverses surfaces telles que des surfaces horizontales, verticales et en pente telles que des sols, des tables basses ou des murs.
- Estimation de la lumière : Permet au téléphone d'estimer les conditions d'éclairage actuelles de l'environnement.

1.2 Comment ARCore fonctionne

Pour résumer, ARCore a deux objectifs : suivre sa position à mesure que les appareils mobiles se déplacent et développer leur propre compréhension du monde réel.

La technologie de suivi des mouvements d'ARCore utilise une caméra de téléphone portable pour identifier les points d'intérêt (appelés points caractéristiques) et suivre le mouvement de ces points dans le temps. En combinant le mouvement de ces points avec les lectures du capteur inertiel du téléphone, ARCore peut déterminer sa position et l'orientation de l'écran à mesure que le téléphone se déplace.

En plus d'identifier les points clés, ARCore détecte également les surfaces planes (telles que les tables ou les sols) et estime l'intensité lumineuse moyenne dans les environs. Ensemble, ces fonctionnalités permettent à ARCore de développer sa propre compréhension du monde qui vous entoure.

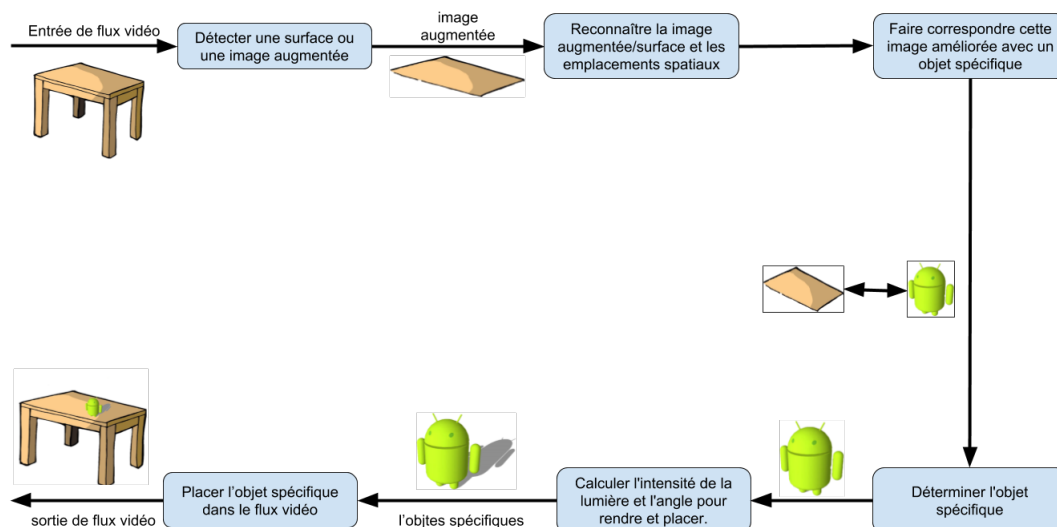


Figure 2 – Processus de reconnaissance d'ARcore

1.2.1 Entrée de flux vidéo

ARCore lance tout d'abord la caméra de l'appareil mobile, qui lui permet d'obtenir un flux vidéo continu et qui fonctionne pour chaque <fragment> du flux vidéo.

ARCore continue de parcourir en boucle chaque <fragment> du flux vidéo (par exemple, en ajoutant un modèle 3D) afin que les modifications correspondantes puissent être observées sur le flux vidéo de sortie (c'est-à-dire l'écran de l'utilisateur).

1.2.2 Reconnaître les images augmentée / surfaces

Lorsque le téléphone se déplace dans le monde, ARCore utilise un processus appelé <odométrie et cartographie concurrentes> pour comprendre où se situe le téléphone par rapport au monde qui l'entoure.

ARCore détecte des caractéristiques visuellement distinctes dans l'image capturée de la caméra, appelées points caractéristiques, et utilise ces points pour calculer son changement d'emplacement. Les informations visuelles sont combinées à des mesures inertielles provenant de l'IMU

de l'appareil pour estimer la pose (position et orientation) de la caméra par rapport au monde à travers le temps.

En alignant la pose de la caméra virtuelle qui restitue le contenu 3D sur celle de la caméra du périphérique fournie par ARCore, les développeurs sont en mesure de restituer le contenu virtuel de la bonne perspective. L'image virtuelle rendue peut être superposée sur l'image obtenue à partir de la caméra du périphérique, ce qui donne l'impression que le contenu virtuel fait partie du monde réel.



Figure 3 – *Le suivi d'ARCore*

1.2.3 Faire correspondre les objets reconnus et les fichiers de ressources

Une fois qu'ARCore a reconnu l'image augmentée ou la surface, ARCore met automatiquement en correspondance le fichier de ressources situé dans la base de données de ressources. ARCore met en correspondance l'objet de fonctionnalité avec le fichier de ressources en vue de la réalité augmentée.

La fonctionnalité de réalité augmentée d'ARCore utilisera le fichier de ressources correspondant pour interagir avec l'objet caractéristique identifié (par exemple, en plaçant la vidéo sur l'objet caractéristique).



Figure 4 – *L'effet d'ARCore*

1.2.4 Compréhension de l'environnement

ARCore améliore constamment sa compréhension de l'environnement réel en détectant des points et des plans.

ARCore recherche les clusters de points de caractéristique qui semblent se trouver sur des surfaces horizontales ou verticales communes, telles que des tableaux ou des murs, et les met à la disposition de votre application en tant que plans. ARCore peut également déterminer les limites de chaque plan et mettre ces informations à la disposition de votre application. Les utilisateurs peuvent utiliser ces informations pour placer des objets virtuels reposant sur des surfaces planes.

Comme ARCore utilise des points de repère pour détecter les plans, les surfaces planes sans texture, telles qu'un mur blanc, risquent de ne pas être détectées correctement.



Figure 5 – Compréhension de l'environnement

1.2.5 Estimation de la lumière

ARCore peut détecter des informations sur l'éclairage de son environnement et vous fournir l'intensité et la correction des couleurs moyennes d'une image de caméra donnée. Ces informations vous permettent d'éclairer vos objets virtuels dans les mêmes conditions que l'environnement qui les entoure, augmentant ainsi le réalisme.

1.2.6 Ancres et traçables

Les poses peuvent changer à mesure qu'ARCore améliore sa compréhension de sa propre position et de son environnement. Lorsque vous souhaitez placer un objet virtuel, vous devez définir une ancre pour vous assurer que ARCore suit la position de l'objet au fil du temps. Souvent, vous créez une ancre en fonction de la pose renvoyée par un test de contact, comme décrit dans Interaction de l'utilisateur.

Le fait que les poses puissent changer signifie qu'ARCore peut mettre à jour la position d'objets environnementaux tels que les plans et les points caractéristiques au fil du temps. Les plans et les points sont un type d'objet spécial appelé traçable. Comme son nom l'indique, il s'agit d'objets que ARCore suivra au fil du temps. Le développeur peut ancrer des objets virtuels à

des objets traçables spécifiques pour s'assurer que la relation entre votre objet virtuel et l'objet traçable reste stable même lorsque le périphérique se déplace.

Cela signifie que si l'utilisateur place une figurine Android virtuelle sur la table, si ARCore ajuste ultérieurement la pose du plan associé à la table, la figurine Android semblera toujours rester au-dessus de la table.

1.2.7 Sortie de flux vidéo

Une fois que ARCore a terminé le traitement de l'objet reconnu et de l'objet placé, ARCore superpose le résultat obtenu dans le flux vidéo et le restitue en continu sur l'écran du appareil mobile, afin que l'utilisateur puisse visualiser le résultat de la réalité augmentée.



Figure 6 – Les résultats obtenus par ARCore

1.3 Interaction utilisateur

ARCore utilise un test de contact pour obtenir les coordonnées (x, y) correspondant à l'écran du téléphone (fournies en appuyant sur une touche ou par toute autre interaction que vous souhaitez que l'application prenne en charge) et projetez un rayon dans le champ de vision de l'appareil photo.

ARCore renverra tous les plans ou les points caractéristiques à travers lesquels le rayon passe et la pose de l'intersection dans un espace réel. Cela permet à l'utilisateur de sélectionner des objets ou d'interagir avec eux dans l'environnement.

ARCore prend en charge plusieurs types d'interactions, tels que le placement d'images fixes, de vidéos, d'animations ou de modèles 3D dans un espace réel. En combinant différentes méthodes d'interaction, ARCore réalise une bonne réalité augmentée.

La grande variété d'interactions de la bibliothèque ARCore permet aux utilisateurs d'en savoir plus sur l'objet identifié par le biais de la réalité augmentée. Mais le seul problème pour le moment est que la couverture des périphériques pris en charge par ARCore est encore faible.

2 La bibliothèque ARToolKit



Figure 7 – Logo ARToolKit

ARToolKit est une bibliothèque de logiciels permettant de créer des applications de réalité augmentée (RA). Ce sont des applications qui impliquent la superposition d'images virtuelles sur le monde réel. L'une des principales difficultés rencontrées dans le développement d'applications de réalité augmentée est le problème du suivi du point de vue des utilisateurs. Pour savoir de quel point de vue l'imagerie virtuelle doit être dessinée, l'application doit savoir où l'utilisateur regarde dans le monde réel. ARToolKit utilise des algorithmes de vision par ordinateur pour résoudre ce problème. Les bibliothèques de suivi vidéo ARToolKit calculent la position et l'orientation réelles de la caméra par rapport aux marqueurs physiques en temps réel. Cela permet de développer facilement une large gamme d'applications en réalité augmentée. Certaines des fonctionnalités d'ARToolKit incluent :

- Suivi de position / orientation de caméra unique.
- Code de suivi utilisant de simples carrés noirs.
- La possibilité d'utiliser n'importe quel modèle de marqueur carré.
- Code d'étalonnage de la caméra facile.
- Assez rapide pour les applications AR en temps réel.
- Distributions SGI IRIX, Linux, MacOS et Windows OS.
- Distribué avec le code source complet.

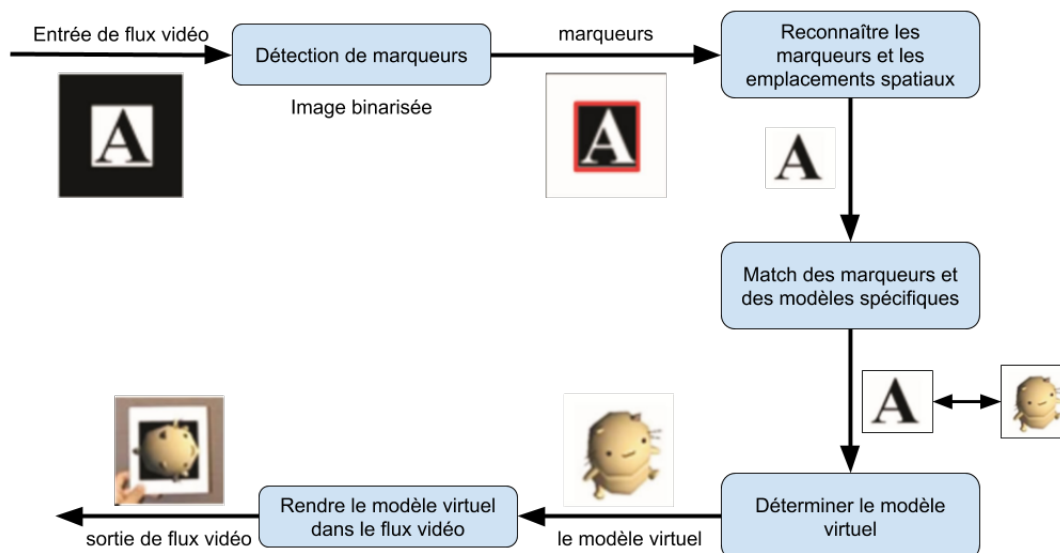


Figure 8 – Processus de reconnaissance d'ARToolKit

ARToolkit prend en charge trois types généraux de technologies de suivi.

- Natural Feature Tracking (NFT) NFT est un standard classique pour la technologie de suivi - prenant en charge toute forme d'image 2d qui n'a pas de contours épitaxiaux clairs et cohérents. Comprend les dessous de verre, les panneaux de signalisation, etc.

- Traditional template square marker Ces marqueurs sont généralement des icônes assez simples avec des limites périphériques noires et blanches fixes. Ce sont tous de bons dessins et pas à l'origine.
- 2D Barcode Markers Ces marqueurs sont prédéfinis dans le SDK et sont souvent profondément optimisés pour une identification rapide et un suivi stable dans diverses conditions d'éclairage. Du fait de son optimisation poussée, il peut être utilisé pour identifier et suivre plusieurs cibles en même temps, et ces marqueurs peuvent être efficaces pour d'autres systèmes de suivi pouvant détecter des défaillances. Cela ressemble plus à une étiquette de cadre ou à un code QR basse résolution qu'à d'autres marqueurs.



Figure 9 – L'effet d'ARToolKit

3 Pourquoi choisir ARCore

En premier lieu, ARCore est une bibliothèque de réalité augmentée développée par Google pour Android, ce qui signifie que pour Android, la meilleure compatibilité sera la bibliothèque de réalité augmentée ARCore. Dans le même temps, ARCore prend en charge la dernière variété de modèles Android et de versions du système Android, ce qui facilite également la mise en œuvre de plusieurs plates-formes.

Deuxièmement, ARCore a la fonction de détection de l'intensité lumineuse, qui assure la génération normale d'objets de réalité augmentée dans différentes conditions d'éclairage et rend également les objets générés plus réalistes.

Dans le même temps, ARCore est plus propice au développement. Pour les programmes intégrant ARCore, toutes les traces sont effectuées sur le périphérique, aucune connexion réseau n'est donc requise. Les images de référence peuvent être mises à jour côté périphérique ou sur le réseau sans appliquer de mises à jour. Lors de son utilisation, ARCore fournit une estimation de la position, de l'orientation et de la taille physique après le suivi de l'image physique. À mesure que les données collectées par ARCore augmentent, ces estimations continuent de s'optimiser, améliorant ainsi la réalité augmentée.

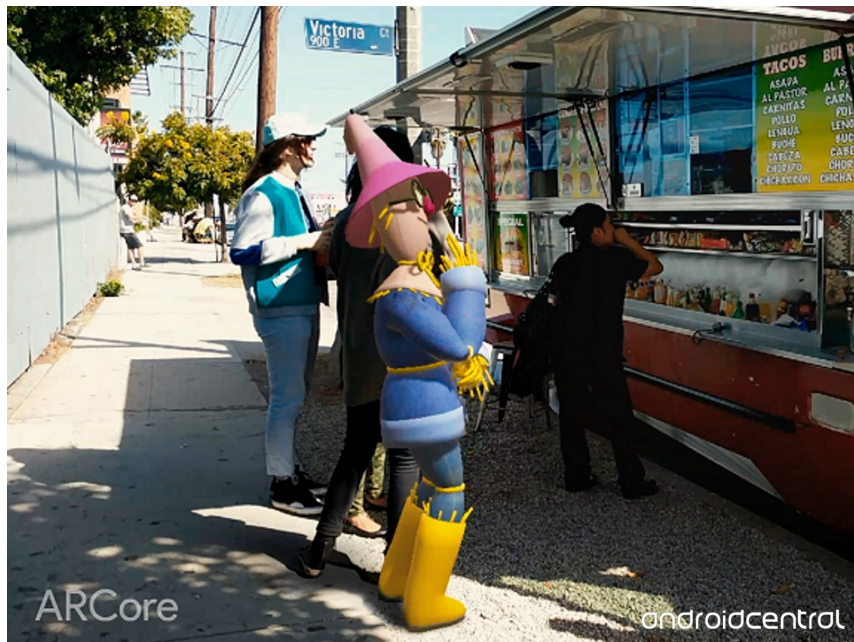


Figure 10 – L'effet de détection de lumière d'ARCore

4

Analyse et conception

1 Image augmentée et base de données d'images augmentée

1.1 Image augmentée

Grâce aux fonctionnalités d'image améliorées d'ARCore, nous pouvons créer des applications AR qui répondent aux images 2D (telles que des affiches ou des emballages de produits) dans un environnement utilisateur.

Lorsque nous fournissons un ensemble d'images de référence, la trace ARCore indique au logiciel l'emplacement physique des images dans la session AR lorsque l'image est détectée dans le champ de vision de la caméra.

- Les images augmentées prennent en charge les formats de fichier PNG et JPEG. Pour les fichiers JPEG, évitez les fortes compressions pour de meilleures performances.
- La détection est uniquement basée sur les points de contraste élevé. Ainsi, les images couleur et noir / blanc sont détectées, qu'une image de référence couleur ou noir / blanc soit utilisée.
- La résolution de l'image doit être d'au moins 300 x 300 pixels.
- L'utilisation d'images haute résolution n'améliore pas les performances.
- Éviter les images avec des fonctionnalités rares.
- Éviter les images avec des fonctionnalités répétitives.
- Une bonne image de référence est difficile à repérer. Nous devons utiliser l'outil `arcoreimg` pour obtenir un score compris entre 0 et 100 pour chaque image. ARCore recommande un score d'au moins 75.

L'évaluation des images augmentées sera effectuée à l'aide de l'outil `arcoreimg`.

1.2 L'outil `arcoreimg`

`arcoreimg` est un outil de ligne de commande qui prend un ensemble d'images de référence et génère un fichier de base de données d'images. Cet outil est disponible dans le SDK ARCore pour Android et s'adresse uniquement aux développeurs Android et Android NDK. Il est intégré aux plug-ins Unity SDK et ARCore Unreal.

L'outil vous permet également de vérifier la qualité de vos images de référence.

1.2.1 Vérifier la qualité de l'image

Une bonne image de référence est difficile à repérer. Nous devons exécuter `arcoreimg eval-img` pour obtenir un score de qualité compris entre 0 et 100 pour chaque image.

```
$ arcoreimg.exe eval-img --input_image_path=dog.png
```

Figure 1 – La commande pour évaluer l'image

ARCore recommande les images avec un score d'au moins 75.

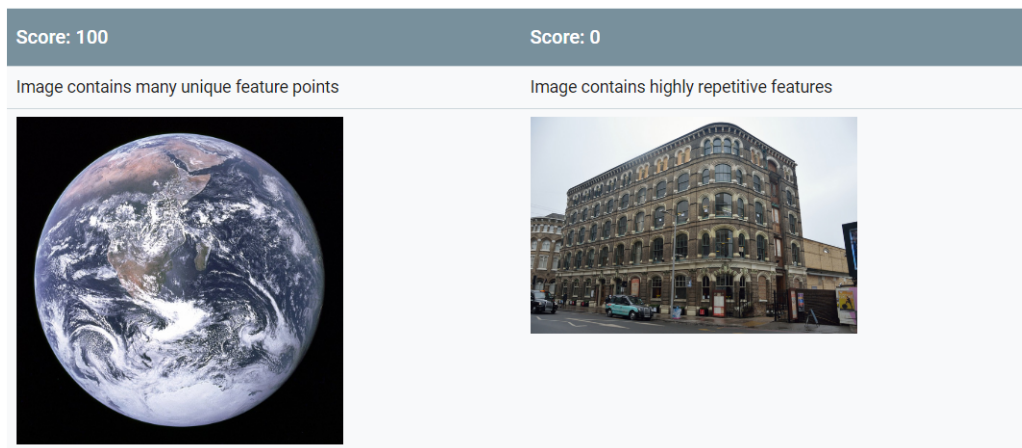


Figure 2 – Norme de notation pour les images augmentées

1.2.2 Créer un fichier de base de données d'images

Nous pouvons fournir des images de référence dans un répertoire ou via un fichier de liste d'images. Cependant, seules les images PNG et JPEG sont prises en charge, et seuls les caractères ASCII sont pris en charge dans les noms de fichiers d'image.

Nous utilisons la commande suivante pour créer une base de données d'images :

```
build-db --input_images_directory=/path/to/images --output_db_path=/path/to/myimages.imgdb
```

Figure 3 – La commande pour créer la base de données d'images

Un nom par défaut est attribué à chaque image en fonction de son chemin. Par exemple, `dog.jpg` a pour nom `/path/to/images/dog.jpg`.

Un index entier de base 0 est également affecté à chaque image, en fonction de l'ordre de tri ASCII des noms de fichiers d'image.

On supposera que toutes les images auront une largeur physique inconnue en mètres.

1.3 Limite de Reconnaissance

Reconnaître et augmenter les images

La liste suivante de fonctionnalités nous aide à évaluer si Images augmentées convient à l'application :

- Chaque base de données d'images peut stocker des informations sur les points caractéristiques pour un maximum de 1000 images de référence.
- ARCore peut suivre jusqu'à 20 images simultanément dans l'environnement, mais il ne peut pas suivre plusieurs instances de la même image.
- L'image physique dans l'environnement doit mesurer au moins 15 cm sur 15 cm et doit être plate (par exemple, pas froissée ou enroulée autour d'une bouteille).
- Une fois suivi, ARCore fournit des estimations de la position, de l'orientation et de la taille physique. Ces estimations sont continuellement affinées au fur et à mesure que ARCore recueille davantage de données.
- ARCore ne peut pas suivre une image en mouvement, mais il peut reprendre le suivi de cette image après son arrêt.
- Tout le suivi se produit sur l'appareil, donc aucune connexion Internet n'est requise. Les images de référence peuvent être mises à jour sur l'appareil ou sur le réseau sans nécessiter de mise à jour d'application.

1.4 Base de données d'images augmentée

Chaque base de données d'images peut stocker des informations pour un maximum de 1000 images.

Il existe deux façons de créer une base de données AugmentedImageDatabase :

- Charge une base de données d'images enregistrée. Ensuite, ajoutez éventuellement d'autres images de référence.
- Crée une nouvelle base de données vide. Ajoutez ensuite les images de référence une à la fois.

Pour créer la base de données d'images :

- La base de données stocke une représentation compressée des images de référence. Chaque image occupe 6Ko.
- Il faut environ 30 ms pour ajouter une image à la base de données au moment de l'exécution. Ajoutez des images sur un thread de travail pour éviter de bloquer le thread de l'interface utilisateur ou, si possible, ajoutez des images au moment de la compilation avec l'outil `arcoreimg`.
- Spécifier la taille physique attendue de l'image. Ces métadonnées améliorent les performances de suivi, en particulier pour les grandes images physiques (plus de 75 cm). Et ne garder pas les images inutilisées dans la base de données car cela aurait un léger impact sur les performances du système.

2 Modélisation logicielle

2.1 Modèle

2.1.1 Reconnaître les images augmentée

La première version du programme doit pouvoir identifier des images améliorées spécifiques avec le support de la bibliothèque ARCore et effectuer des interactions simples pour vérifier le succès de la reconnaissance.

Dans cette version, nous n'avons besoin que d'une interface principale <MainActivity>, qui tirera parti de la caméra du périphérique mobile, gardera la caméra allumée et gardera une trace de l'espace réel.

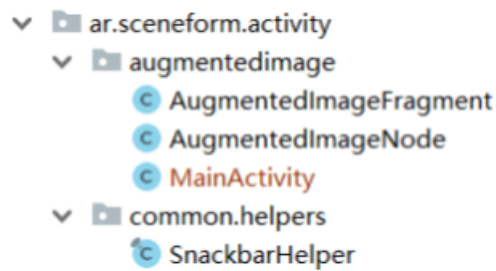


Figure 4 – La structure d'application version 1.0

Le programme contient maintenant quatre fichiers java :

- AugmentedImageFragment.java :
Ce fichier étend ArFragment pour personnaliser la configuration de session ARCore afin d'inclure des images augmentées.
- AugmentedImageNode.java :
Ce fichier définit un nœud pour le rendu d'une image augmentée. L'image est encadrée en plaçant le cadre d'image virtuel aux coins de l'image augmentée pouvant faire l'objet d'un suivi.
- MainActivity.java :
Ce fichier est l'interface principale du programme, qui définit la disposition du composant et configure l'écouteur de suivi, définit le comportement de l'interaction et charge la base de données d'images augmentée.
- SnackbarHelper.java
Ce fichier gère l'exemple de snackbar. Cache le code général Android et expose des méthodes plus simples.

2.1.2 Reconnaître d'objet 3D

La deuxième version du programme commence à permettre à l'utilisateur de reconnaître les objets 3D, mais ne reconnaît qu'un objet 3D et contient toujours une interface principale distincte pour le suivi d'objet reconnaissables. De plus, le programme utilise le nouveau <Fragment> personnalisé.

Pour les fichiers de ressources, le programme ajoutera des images améliorées d'angles différents, de directions différentes et de distances différentes du même objet 3D. Le programme suivra en permanence ces images augmentées pour permettre de suivre l'objet 3D.

Une fois que l'objet 3D correspondant est suivi, le programme place un modèle 3D sur l'objet, ce qui nous aide à vérifier le succès du suivi du programme.

La structure du programme est la suivante :

- MainActivity.java :
Ce fichier est l'interface principale du programme, qui définit la disposition du composant et configure l'écouteur de suivi, définit le comportement de l'interaction et charge la base de données d'images augmentée.
- CustomARFragment.java
Ce fichier étend ArFragment pour personnaliser la configuration de session ARCore afin d'inclure des images augmentées.

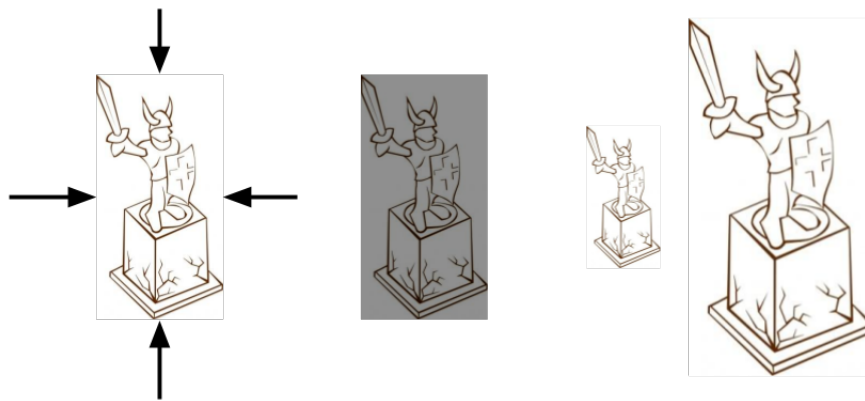


Figure 5 – La façon de prendre les images améliorée

2.1.3 Reconnaître plusieurs objets 3D et ajouter des interactions

La troisième version du programme implémentera le reconnaissance de plusieurs objets 3D et affichera différentes interactions lors du suivi de différentes directions du même objet (par exemple, affichage de différentes vidéos, différents modèles 3D, différentes informations). Cette version ajoutera plus d'images augmentée d'objets 3D et différents fichiers de ressources pour obtenir les fonctionnalités souhaitées.

Cette version ajoutera également une interface simple, telle que le menu principal. En même temps, une interface sera ajoutée pour permettre à l'utilisateur de parcourir les objets 3D reconnaissable stockés dans la base de données. L'interface principale fonctionne toujours pour accepter les flux vidéo, suivre et interagir.

La structure du programme est la suivante :

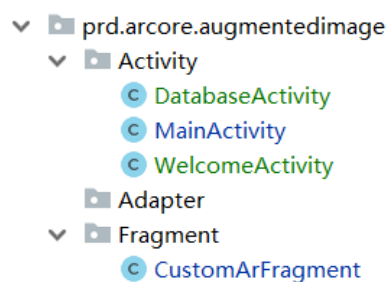


Figure 6 – La structure d'application version 3.0

- DatabaseActivity.java : Cette interface est utilisée pour permettre à l'utilisateur de visualiser les objets reconnaissables stockés dans la base de données. Pour chaque objet reconnaissable, l'interface contiendra une image améliorée de celui-ci et les descriptions textuelles associées.
- MainActivity.java : Ce fichier est l'interface principale du programme, qui définit la disposition du composant et configure l'écouteur de suivi, définit le comportement de l'interaction et charge la base de données d'images augmentée.
- WelcomeActivity.java : Cette interface est le menu principal du programme, qui définit la présentation du menu principal et les événements de réponse de la page de saut.

- CustomArFragment.java : Ce fichier étend ArFragment pour personnaliser la configuration de session ARCore afin d'inclure des images augmentées.

5

Mise en oeuvre

1 Mise en oeuvre du semestre 9

Au semestre 9, l'objectif principal est d'apprendre et de comprendre les principes et l'utilisation de la bibliothèque ARCore et des outils associés. Étant donné que je n'avais pas d'appareil à tester au cours du semestre 9, je me suis concentré sur l'utilisation de l'outil `arcoreimg` :

- prendre les images augmentée
- faire l'évaluation de les images augmentée
- faire la création et l'utilisation de bases de données d'images augmentée

et l'utilisation des outils Sceneform.

1.1 Utiliser l'outil `arcoreimg`

1.1.1 Prendre les images augmentée

Selon les idées mentionnées précédemment, j'ai commencé des recherches sur les images augmentée sous S9, qui est un facteur clé de la reconnaissance ARCore, c'est pourquoi j'ai réfléchi à la méthode que j'ai choisie.

Étant donné que l'application doit être utilisée pour des expositions d'art telles que des sculptures, il faut tenir compte du fait que les images augmentée dans différentes directions seront différentes. Pour une même sculpture, nous devons utiliser au moins 8 directions comme angle de prise de vue et tenir compte également des angles différents des téléphones portables détenus par différents utilisateurs. Nous considérons ici trois angles différents, donc pour chaque objet 3D à identifier, Au moins 24 images augmentée sont requises.

Pour le même objet identifiable, nous utilisons des images améliorées prises dans 8 directions différentes. Les 8 directions sont : avant, arrière, gauche, droite, avant gauche, arrière gauche, avant droite, arrière droite.

Pour les angles, nous utilisons 3 angles différents. Les trois angles de prise de vue sont : de bas en haut (vue vers le haut), angle de vue parallèle, de haut en bas (vue de dessus)

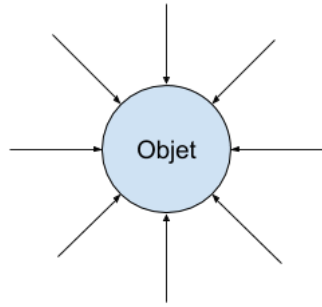


Figure 1 – La façon de capturer une image augmentée

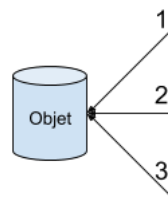


Figure 2 – Différents angles pour image augmentée

1.1.2 faire l'évaluation de les images augmentée

Comme ARCore spécifie les critères de l'image augmentée utilisée, nous devons d'abord les évaluer pour toutes les images augmentée, à l'aide de l'outil arcoiling. Pour chaque image augmentée, arcoeimg donnera le score correspondant (La plage de score est comprise entre 0 et 100). ARCore a suggéré que le programme utilise des images augmentée avec plus de 75 points. Ainsi, après avoir obtenu les images augmentée, j'ai commencé à les évaluer à l'aide de l'outil acoreimg.

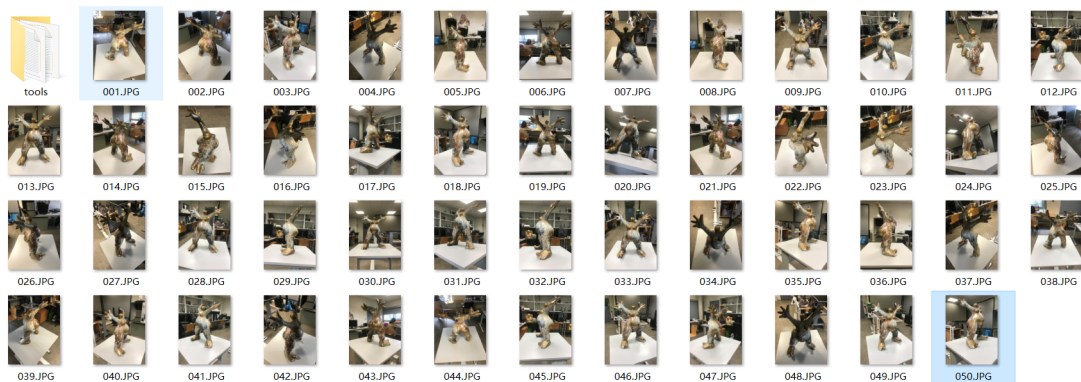


Figure 3 – Les images augmentées

arcoring est un outil de ligne de commande, chaque évaluation nécessite de taper une commande. Quand il besoin de évaluer un grand nombre d'images, il est très difficile de l'exécuter sur la ligne de commande. Par conséquent, j'ai utilisé un script batch pour faire l'évaluation tous les images.

En scoring, nous avons constaté que l'image caractéristique préférée d'ARCore est une image capable de mettre en évidence les caractéristiques de l'objet identifié, ce qui signifie que les points caractéristiques de l'image doivent être autant que possible et que les objets identifiés

```

1 @echo off
2
3 @set num=1
4
5 :loop
6 @if %num% geq 10 goto loop2
7 .\arccoreimg.exe eval-img --input_image_path=D:\ARCORE\DatabaseWoman\00%num%.jpg
8 @echo Finished evaluation image %num%
9 set /a num+=1
10
11 @goto loop
12
13 :loop2
14 .\arccoreimg.exe eval-img --input_image_path=D:\ARCORE\DatabaseWoman\0%num%.jpg
15 @if %num% equ 51 goto end
16 @echo Finished evaluation image %num%
17 set /a num+=1
18 @goto loop2
19
20 :end
21 @echo Finished all 50 images
22 pause
23

```

Figure 4 – Bat script utilisant l'outil arccoreimg

doivent présenter de grandes différences d'arrière-plan. Lorsque la couleur de l'objet identifié est plus proche de l'arrière-plan ou que le contour de l'objet identifié n'est pas évident, l'outil arcieimg attribue un score très faible à l'image (0-25 points). Inversement, lorsque l'objet de la caractéristique est très important ou très différent de l'environnement environnant, le score est élevé (75-100 points).

100	
Finished evaluation image 1	Finished evaluation image 13
50	90
Finished evaluation image 2	Finished evaluation image 14
100	100
Finished evaluation image 3	Finished evaluation image 15
85	60
Finished evaluation image 4	Finished evaluation image 16
90	50
Finished evaluation image 5	Finished evaluation image 17
100	100
Finished evaluation image 6	Finished evaluation image 18
85	95
Finished evaluation image 7	Finished evaluation image 19
80	100
Finished evaluation image 8	Finished evaluation image 20
95	25
Finished evaluation image 9	Finished evaluation image 21
100	100
Finished evaluation image 10	Finished evaluation image 22
75	100
Finished evaluation image 11	Finished evaluation image 23
100	95
Finished evaluation image 12	Finished evaluation image 24
100	

Figure 5 – Résultats d'évaluation pour les images

1.1.3 Faire la création et l'utilisation de bases de données d'images augmentée

Une fois l'évaluation de toutes les images augmentée terminée, nous sélectionnons toutes les images augmentée dont le score est supérieur à 75 et commençons à utiliser ces images pour la création de la bases de données d'images augmentée.

Il existe deux méthodes de créer une base de données, l'une par la ligne de commande et l'autre par code à l'aide de la méthode AugmentedImageDatabase().

Lors du processus de création d'une base de données d'images augmentée via la ligne de commande, j'ai constaté que, sous Windows, la taille de chaque base de données d'images augmentée créée est de 0 Ko, ce qui signifie que la base de données n'a pas été créée avec succès. Après plusieurs tentatives, la situation est toujours la même. Je pense que cela peut être un problème avec arcoring sous Windows, je vais donc choisir la deuxième méthode, qui consiste à utiliser le code pour créer une base de données d'images augmentée dans le programme.

```
AugmentedImageDatabase augmentedImageDatabase;
augmentedImageDatabase = new AugmentedImageDatabase(session);
```

Figure 6 – La méthode `AugmentedImageDatabase()`

1.2 Utiliser l'outil Sceneform

Sceneform facilite le rendu de scènes 3D réalistes dans les applications AR et non AR. Il comprend :

- API de graphique de scène de haut niveau
- Un rendu physique réaliste fourni par Filament
- Un plugin Android Studio pour importer, afficher et créer des éléments 3D

Dans mon projet, j'ai utilisé l'outil Sceneform pour importer et créer des modèles 3D dans des fichiers de ressources, ainsi que pour restituer des scènes.

1.2.1 Importer et créer des modèles 3D

Pour importer et utiliser des modèles 3D dans Sceneform, il faut vérifier que le plug-in Google Sceneform Tools (Beta) est installé ou pas.

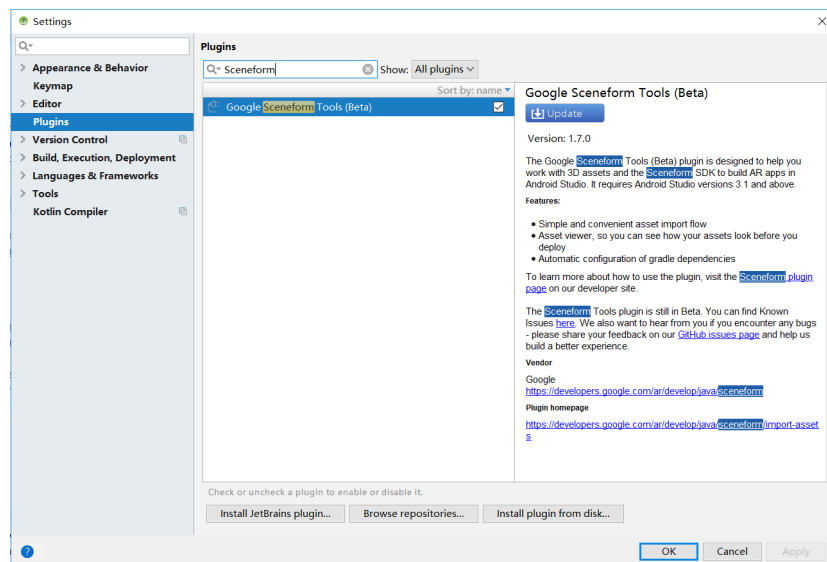


Figure 7 – Le plug-in Google Sceneform Tools (Beta)

Sceneform prend en charge les ressources 3D dans les formats suivants : *OBJ, *glTF (animations non prises en charge) *FBX (avec ou sans animations).

Pour importer un nouvel élément 3D :

- Vérifiez que le dossier d'applications de votre projet contient un dossier `sampledata`.

- Cliquer avec le bouton droit sur le dossier de l'application dans la fenêtre Projet, puis sélectionne Nouveau > Répertoire de données exemple pour créer le dossier.
- Le dossier sampledata fait partie de votre projet Android Studio, mais son contenu ne sera pas inclus dans APK.
- Copier le fichier d'actif source du modèle 3D (* .obj, * .fbx ou * .gltf) et toutes ses dépendances (* .mtl, * .bin, * .png, * .jpg, etc.) dans le fichier sampledata. dossier.
- Ne copier pas ces fichiers source dans les actifs ou le dossier res de votre projet pour éviter d'être inclus dans APK.
- Cliquer avec le bouton droit sur la ressource source du modèle 3D et sélectionnez Importer une ressource Sceneform pour lancer le processus d'importation.

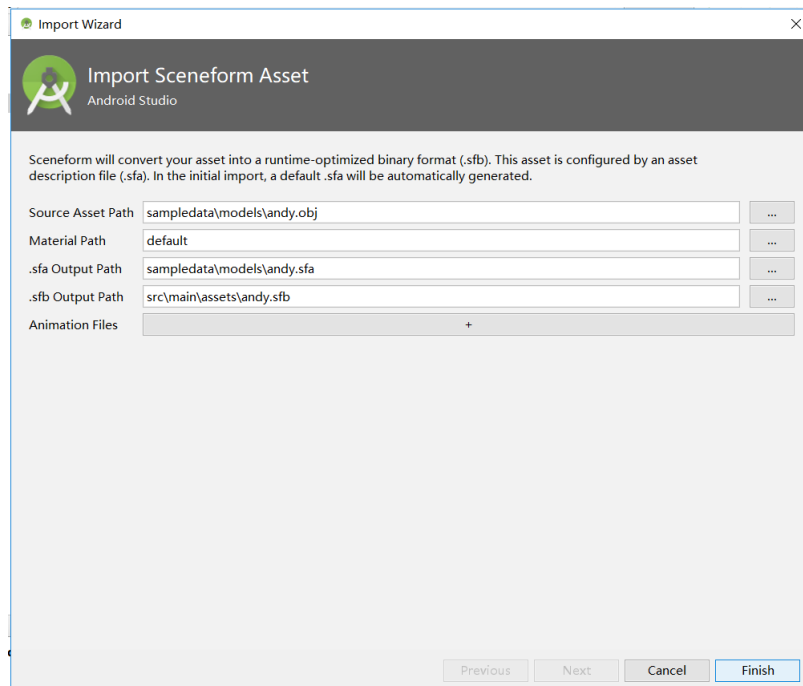


Figure 8 – Import Sceneform Asset

Ensuite, nous pouvons voir le modèle 3D créé par le fichier .obj, qui peut être utilisé comme fichier de ressources et utilisé ultérieurement lorsque nous écrivons un comportement interactif.

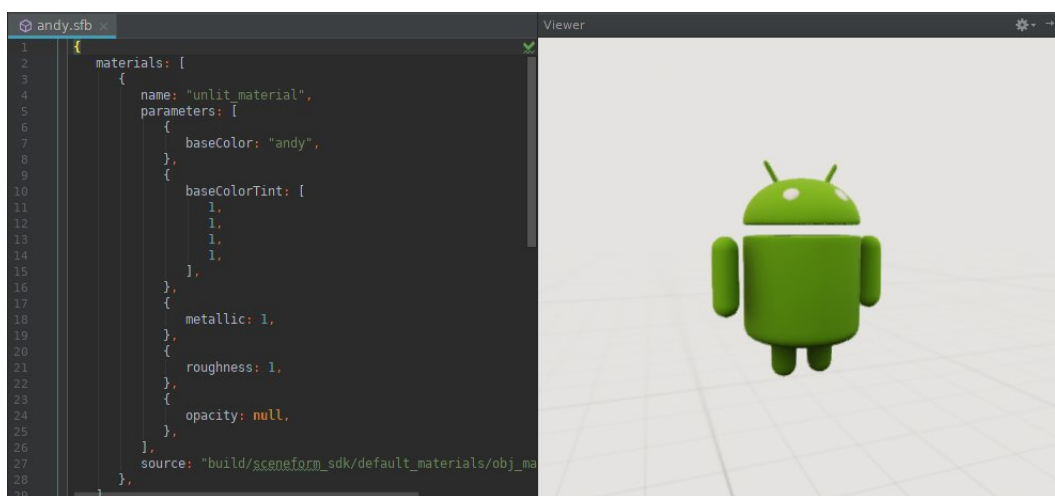


Figure 9 – Afficher le ressource du modèle 3D

2 Mise en oeuvre du semestre 10

Au cours du semestre 10, mon objectif principal était d'écrire le programme Android de reconnaissance d'objets en 3D, que j'ai divisé en trois parties, à savoir l'écriture de trois versions de l'application. La première version implémente la reconnaissance des images augmentées, la seconde version permet la reconnaissance d'objets 3D individuels et la troisième version permet la reconnaissance de plusieurs objets 3D et l'ajout d'interactions supplémentaires lors de la création de menus.

2.1 Reconnaître les images augmentées

Dans cette version, notre objectif est d'obtenir la reconnaissance et l'interaction d'images augmentées qui ont été évaluées et importées à l'avance.

J'ai d'abord besoin de charger la base de données d'images améliorée dans le programme. J'ai utilisé la méthode `SetupAugmentedImageDb()`.

```
private boolean setupAugmentedImageDatabase(Config config, Session session) {
    AugmentedImageDatabase augmentedImageDatabase;

    AssetManager assetManager = getContext() != null ? getContext().getAssets() : null;
    if (assetManager == null) {
        Log.e(TAG, msg: "Context is null, cannot initialize image database.");
        return false;
    }
}
```

Figure 10 – La méthode `SetupAugmentedImageDb()`

Après avoir créé la base de données, j'ai ajouté l'image augmentées à la base de données. Je l'ai implémentée à l'aide de `augmentedImageDatabase.addImage()`.

```
augmentedImageDatabase = new AugmentedImageDatabase(session);
augmentedImageDatabase.addImage(DEFAULT_IMAGE_NAME, augmentedImageBitmap);
```

Figure 11 – La méthode `augmentedImageDatabase.addImage()`

J'ai ensuite créé `loadAugmentedImageBitmap()` Permet au programme de lire la base de données d'images augmentées.

```
private Bitmap loadAugmentedImageBitmap(AssetManager assetManager) {
    try (InputStream is = assetManager.open(DEFAULT_IMAGE_NAME)) {
        return BitmapFactory.decodeStream(is);
    } catch (IOException e) {
        Log.e(TAG, msg: "IO exception loading augmented image bitmap.", e);
    }
    return null;
}
```

Figure 12 – La méthode `loadAugmentedImageBitmap()`

Enfin, dans l'interface principale, j'utilise la méthode `onUpdateFrame()` pour suivre en continu l'image augmentée dans le flux vidéo.

Les résultats d'exécution de cette version du programme sont les suivants : Après avoir suivi l'image augmentée, un modèle 3D du robot Android sera placé sur l'image augmentée.

```
private void onUpdateFrame(FrameTime frameTime) {
    Frame frame = arFragment.getArSceneView().getArFrame();

    // If there is no frame or ARCore is not tracking yet, just return.
    if (frame == null || frame.getCamera().getTrackingState() != TrackingState.TRACKING) {
        return;
    }

    Collection<AugmentedImage> updatedAugmentedImages =
        frame.getUpdatedTrackables(AugmentedImage.class);
    for (AugmentedImage augmentedImage : updatedAugmentedImages) {
        switch (augmentedImage.getTrackingState()) {
```

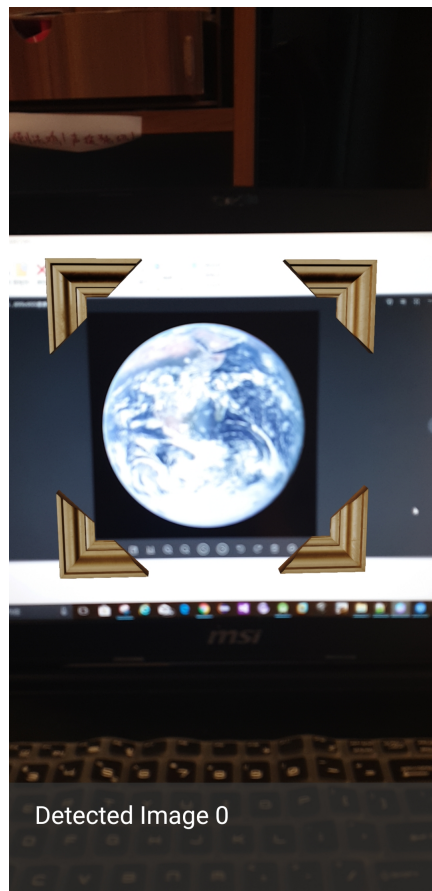
Figure 13 – La méthode `onUpdateFrame()`

Figure 14 – Le programme Version 1.0

2.2 Reconnaître l'objet 3D

Dans cette version, l'objet reconnu est remplacé par une image 2D par un objet 3D et, comme je l'ai déjà dit, je vais obtenir plusieurs images augmentée du même objet 3D et les enregistrer dans la base de données. Je dois également modifier certaines méthodes en conséquence.

Pour la méthode `SetupAugmentedImageDb()`, j'ai utilisé des boucles pour ajouter des images augmentée, et la base de données d'images augmentée résultante contiendrait différentes images augmentée du même objet 3D.

Dans le même temps, je redéfinit la méthode `onUpdateFrame()` afin que le programme puisse suivre simultanément toutes les images augmentée, c'est-à-dire qu'il puisse reconnaître et interagir avec l'objet dans différentes directions ou angles.

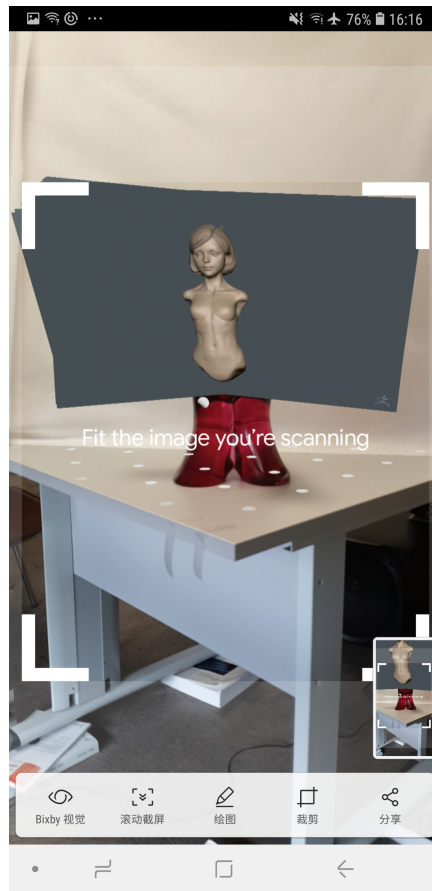


Figure 15 – Le programme Version 2.0

D'après les résultats, nous pouvons constater que le programme a réussi à tracer l'objet 3D (statue rouge) et à placer l'image statique dans le fichier de ressources sur l'objet, ce qui montre que la reconnaissance de l'objet 3D a réussi.

2.3 Reconnaître plusieurs objets 3D et ajouter des interactions

Dans cette version, j'ai trois tâches :

- Ajouter plus d'objets 3D reconnaissables (ajouter plus d'images augmentée)
- Ajouter plus d'interactions (ajouter une vidéo interactive)
- Ajouter une interface de programme (menu principal et interface d'affichage pour les objets identifiables)

2.3.1 Ajouter plus d'objets 3D reconnaissables

Pour que le programme reconnaisse plus d'objets 3D, je ajoute plus d'images augmentée à la base de données, tout en identifiant différents objets 3D, il me faut classifier les images augmentée, ainsi que modifier la fonction pour ajouter les images augmentée à la base de donnée.

Pour l'implémentation de la classification, j'ai utilisé la méthode de nommage du bitmap : pour les images augmentée du même objet 3D, je leur ai donné le même nom, tel que "woman", "baby", etc. Lorsque le programme suit ces images augmentée, il suivra également les différents noms. De cette manière, le programme complète la classification de ces images augmentée.

```

public boolean setupAugmentedImagesDb(Config config, Session session) {
    AugmentedImageDatabase augmentedImageDatabase;
    augmentedImageDatabase = new AugmentedImageDatabase(session);
    String name;
    for(int i = 1; i<49;i++){
        Bitmap bitmap = loadAugmentedImage(i);
        if (bitmap == null) {
            return false;
        }
        else if(i<8){
            augmentedImageDatabase.addImage(S: "woman1", bitmap);
        }
        else if(i>=8 && i <=16){
            augmentedImageDatabase.addImage(S: "woman2", bitmap);
        }
        else if(i > 16 && i<=22){
            augmentedImageDatabase.addImage(S: "baby", bitmap);
        }
    }
}

```

Figure 16 – Modifier la méthode SetupAugmentedImageDb()

```

@RequiresApi(api = Build.VERSION_CODES.N)
private void onUpdateFrame(FrameTime frameTime) {
    Frame frame = arFragment.getArSceneView().getArFrame();

    Collection<AugmentedImage> augmentedImages = frame.getUpdatedTrackables(AugmentedImage.class);
    for (AugmentedImage augmentedImage : augmentedImages) {
        if (augmentedImage.getTrackingState() == TrackingState.TRACKING) {
            if (augmentedImage.getName().equals("woman1") && shouldAddModel) {
                //placeObject(arFragment, augmentedImage.createAnchor(augmentedImage.getCenterPose()), Uri.parse("andy.sfb"));
                videoPlayer(context: this, augmentedImage, AugmentedImageType: "woman1");
                shouldAddModel = true;
            }
            else if (augmentedImage.getName().equals("woman2") && shouldAddModel) {
                placeObject(arFragment, augmentedImage.createAnchor(augmentedImage.getCenterPose()), Uri.parse("woman.sfb"));
                videoPlayer(context: this, augmentedImage, AugmentedImageType: "woman2");
                shouldAddModel = true;
            }
            else if (augmentedImage.getName().equals("baby") && shouldAddModel) {
                placeObject(arFragment, augmentedImage.createAnchor(augmentedImage.getCenterPose()), Uri.parse("andy.sfb"));
                videoPlayer(context: this, augmentedImage, AugmentedImageType: "baby");
                shouldAddModel = true;
            }
        }
    }
}

```

Figure 17 – Modifier la méthode onUpdateFrame()

2.3.2 Ajouter plus d'interactions

Pour ajouter plus d'interaction, j'ai choisi de superposer la vidéo sur l'objet reconnaissable, ce qui donnera à l'utilisateur une compréhension plus intuitive de l'objet identifiable. Afin de réaliser l'interaction de la superposition vidéo, j'ai utilisé ChromeKey.

De même, j'ai ajouté une nouvelle méthode videoPlayer(), qui place la vidéo dans le fichier de ressources devant l'objet reconnu de manière parallèle pour obtenir un comportement interactif et superposer la vidéo.

Dans le choix de la vidéo, j'utilise des fichiers vidéo au format MP4. Ce fichier est de petite taille, facile à stocker et en haute qualité. De nombreuses vidéos sur écran vert sont également au format MP4. Dans ce programme, je stocke différents fichiers ressources. Le fichier vidéo se trouve dans le dossier <raw> et la méthode videoPlayer() associera différents objets 3D à différentes vidéos.

```

@RequiresApi(api = Build.VERSION_CODES.N)
public void videoPlayer(Context context, AugmentedImage augmentedImage, String AugmentedImageType) {
    // Create an ExternalTexture for displaying the contents of the video.
    ExternalTexture texture = new ExternalTexture();
    if (AugmentedImageType == "babydrink") {
        mediaPlayer = MediaPlayer.create(context, this, R.raw.baby);
    } else if (AugmentedImageType == "woman1") {
        mediaPlayer = MediaPlayer.create(context, this, R.raw.fireworks3);
    } else if (AugmentedImageType == "chocolate") {
        mediaPlayer = MediaPlayer.create(context, this, R.raw.chocolate);
    } else if (AugmentedImageType == "womanfront") {
        mediaPlayer = MediaPlayer.create(context, this, R.raw.womanfront);
    } else if (AugmentedImageType == "womanright") {
        mediaPlayer = MediaPlayer.create(context, this, R.raw.womanright);
    }
}

```

Figure 18 – La méthode videoPlayer()

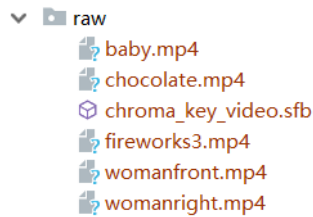


Figure 19 – Le dossier <raw>

2.3.3 Ajouter une interface de programme

Enfin, j'ai ajouté un menu principal au programme et une autre interface pour présenter à l'utilisateur des objets reconnaissable. Le menu principal est utilisé pour la navigation.

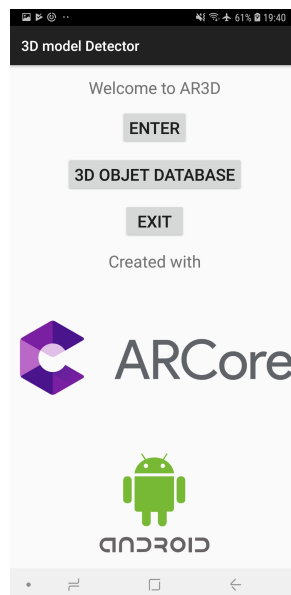


Figure 20 – Le menu principal

Le programme peut éventuellement suivre différents objets 3D et afficher différentes vidéos dans différentes directions.

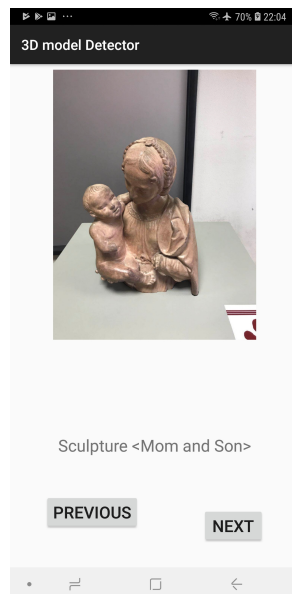


Figure 21 – *L'interface pour présenter les objets reconnaissable*

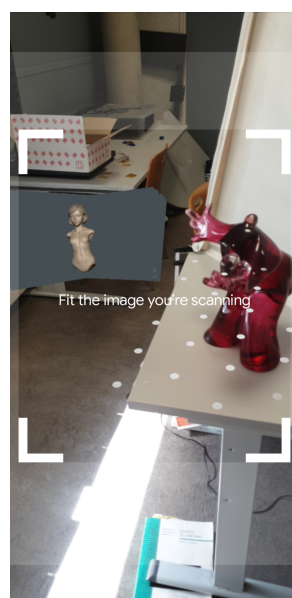


Figure 22 – *Effet de reconnaissance finale du programme 1*

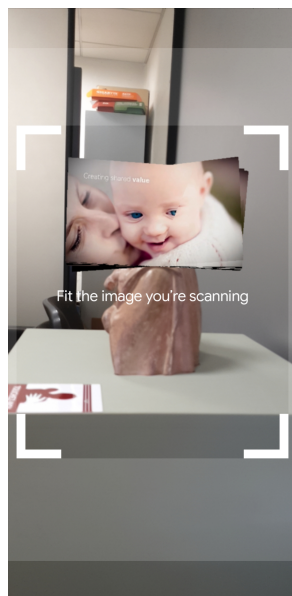


Figure 23 – *Effet de reconnaissance finale du programme 2*

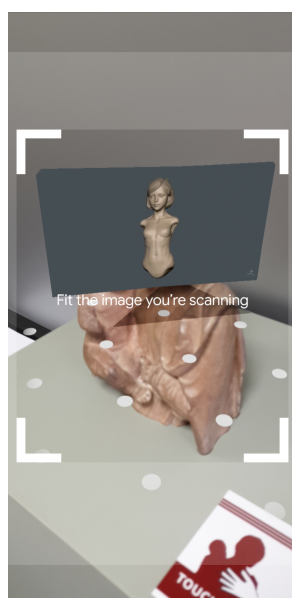


Figure 24 – *Effet de reconnaissance finale du programme 3*

6

Bilan et conclusion

1 Bilan du semestre 9

Dans la première partie de PRD, mon travail principal consistait à explorer et analyser ARCore, à tester les performances et les propriétés d'ARCore, et à planifier les fonctions et les méthodes que le programme devrait mettre en œuvre.

Parce que la bibliothèque de réalité augmentée utilisée dans ce projet, ARCore, est un outil de réalité augmentée nouvellement développé, il existe peu de documents de support sur le réseau, ce qui limite notre analyse des logiciels existants développés à l'aide de la bibliothèque ARCore. Nous devons donc faire plus de tests pour comprendre le fonctionnement d'ARCore.

Lors de la première partie du travail, j'ai rencontré des problèmes matériels, car les appareils pris en charge par ARCore n'incluaient pas tous les modèles Android. J'ai donc attendu l'arrivée de nouveaux appareils mobiles. Le temps de test a été légèrement retardé.

L'analyse de la première partie m'a permis d'avoir un plan clair pour les travaux de développement ultérieurs et de diviser mon travail en parties détaillées. Dans la deuxième partie du PRD, je commencerai l'élaboration du programme. La première chose à faire est d'utiliser ARCore pour la reconnaissance d'objets 3D.

2 Bilan du semestre 10

Pour la deuxième partie de mon PRD, mon travail principal consiste à écrire des applications Android et à utiliser les outils et méthodes liés à ARCore, ainsi qu'à obtenir la reconnaissance des objets 3D.

J'ai divisé le travail de la deuxième partie de PRD en trois petites parties, à savoir trois versions de l'application, la première version identifiant l'image augmentée, la deuxième version identifiant un seul objet 3D et rejoignant le comportement interactif, la troisième version. Il s'agit du menu principal et d'autres interfaces permettant d'identifier plusieurs objets 3D, d'ajouter des interactions vidéo et d'ajouter des interfaces.

Lors de la rédaction du programme, la principale difficulté que j'ai rencontrée est la création de la base de données d'images augmentée. Selon la documentation officielle d'ARCore, ARCore peut généralement utiliser l'outil `arc4img` pour créer une base de données d'images augmentée

sur la ligne de commande, mais sur mon ordinateur. J'ai essayé plusieurs fois sans succès et j'ai également effectué des recherches en ligne. J'ai finalement choisi de modifier le mode de création de la base de données. Cette fois, j'ai réussi à créer une base de données d'images augmentée.

3 Reste à faire

Pour ce projet, la fonction de reconnaissance, de suivi et d'interaction avec plusieurs objets 3D a été réalisée et le programme dispose déjà d'une interface de navigation simple.

De même, le programme présente encore des imperfections et certains travaux restent à faire :

- Les effets de reconnaissance sont parfois affectés par les objets d'arrière-plan, la luminosité
- Le temps de démarrage est légèrement plus long (environ 20 secondes)
- L'interface est encore relativement simple

Les prochains travaux de développement devraient se concentrer sur ces aspects et les performances du logiciel doivent encore être améliorées aussi.

4 Test

Dans ce projet, j'ai utilisé le cadre de test automatisé d'Android, Espresso.

L'outil Espresso Test Recorder permet au développeur de créer des tests d'interface utilisateur pour les applications. En enregistrant un scénario de test, je peux enregistrer les interactions avec un appareil et ajouter des assertions pour vérifier les éléments de l'interface utilisateur dans des instantanés particuliers de l'application. Espresso Test Recorder prend ensuite l'enregistrement sauvegardé et génère automatiquement un test d'interface utilisateur correspondant que je peux exécuter pour tester l'application.

Espresso Test Recorder écrit des tests basés sur le framework Espresso Testing, une API dans AndroidX Test. L'API Espresso nous encourage à créer des tests d'interface utilisateur concis et fiables basés sur les actions de l'utilisateur. En énonçant des attentes, des interactions et des assertions sans accéder directement aux activités et aux vues de l'application sous-jacente, cette structure évite les erreurs de test et optimise la vitesse d'exécution.

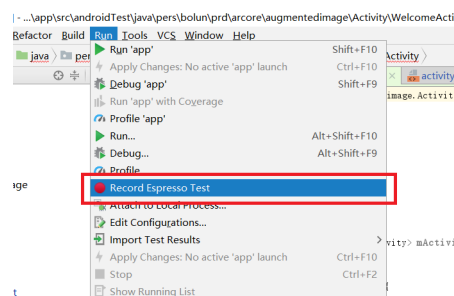


Figure 1 – Le menu <Record Espresso Test>

Les interactions enregistrées apparaîtront dans le panneau principal de la fenêtre <Record Your Test>. Lorsque l'application exécute le test, le test Espresso essaiera d'exécuter ces actions dans le même ordre.

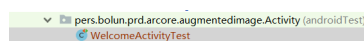


Figure 2 – Le fichier <WelcomeActivityTest>

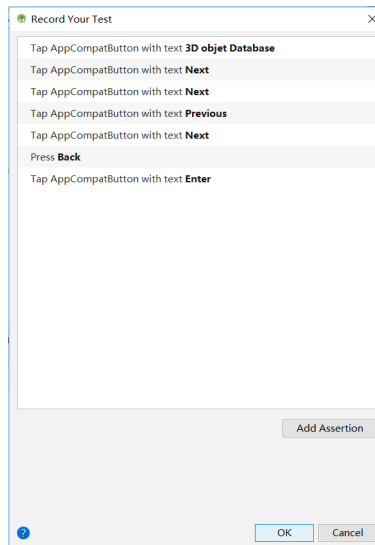


Figure 3 – La fenêtre <Record Your Test>

5 Conclusion

Dans le projet PRD de deux semestres, j'ai beaucoup appris de ce projet, principalement sur deux aspects :

- Tout d'abord, j'ai appris à utiliser la bibliothèque ARCore de Google. C'est la première fois que je travaille sur la programmation en réalité augmentée. Pour ARCore et les outils connexes (arcoreimg et Sceneform), j'ai une compréhension et un apprentissage approfondis de ce projet PRD. . J'ai maîtrisé les bases de son utilisation et appris à l'appliquer.
- Deuxièmement, j'ai appris la programmation de l'application Android. De même, le projet PRD est la première fois que je travaille sur la programmation de l'application Android. Durant ce projet, j'ai beaucoup travaillé d'apprentissage et de consultation pour comprendre le programme Android. Principe de fonctionnement, structure grammaticale, cadre, etc.

Pour la gestion du projet, au cours des deux semestres, j'ai divisé mon travail en détail et appris à utiliser le système de gestion de version de logiciel, dans lequel j'ai utilisé GitHub + GitKraken pour gérer mon logiciel. La version et le diagramme de Gantt ont été utilisés pour planifier les travaux, ce qui a permis de faire avancer mon projet de manière ordonnée.

Enfin, je suis très reconnaissant à Monsieur GILLES VENTURINI, Madame MARIANNE BENNET et tous les professeurs qui m'ont aidé au PRD.

Annexes

A

Planification

1 Aperçu de gestion de projet

Le diagramme de Gantt pour la planification de ce projet est comme Figure1.

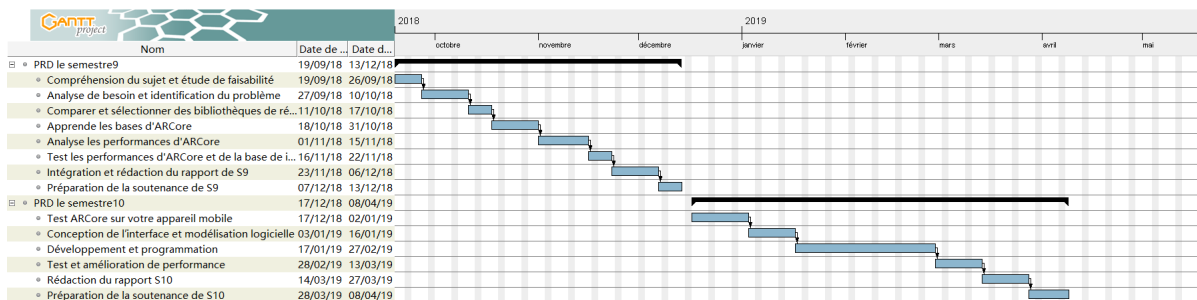


Figure 1 – Le diagramme de Gantt

Pour le S9, le projet commence le 19/09/2018 par la compréhension du sujet et finit le 13/12/2018 par la soutenance de S9.

Pour le S10, le projet commence le 17/12/2018 en Test ARCore sur votre appareil mobile et le projet finit le 05/04/2019 par la soutenance de S10. En moyen, chaque tâche dure 1 ou 2 semaines, mais pour la programmation et le test, ils prendront environ 5 ou 6 semaines.

2 Aperçu de gestion de projet

1.Compréhension du sujet et étude de faisabilité

Date de début : le 19 septembre 2018;

Date de fin : le 26 septembre 2018;


Durée : 7 jours;

2.Analyse de besoin et identification du problème

Date de début : le 27 septembre 2018;

Date de fin : le 10 octobre 2018;

Durée : 14 jours;



Nom	Date de ...	Date d...
PRD le semestre9	19/09/18	13/12/18
• Compréhension du sujet et étude de faisabilité	19/09/18	26/09/18
• Analyse de besoin et identification du problème	27/09/18	10/10/18
• Comparer et sélectionner des bibliothèques de ré...	11/10/18	17/10/18
• Apprendre les bases d'ARCore	18/10/18	31/10/18
• Analyse les performances d'ARCore	01/11/18	15/11/18
• Test les performances d'ARCore et de la base de i...	16/11/18	22/11/18
• Intégration et rédaction du rapport de S9	23/11/18	06/12/18
• Préparation de la soutenance de S9	07/12/18	13/12/18
PRD le semestre10	17/12/18	08/04/19
• Test ARCore sur votre appareil mobile	17/12/18	02/01/19
• Conception de l'interface et modélisation logicielle	03/01/19	16/01/19
• Développement et programmation	17/01/19	27/02/19
• Test et amélioration de performance	28/02/19	13/03/19
• Rédaction du rapport S10	14/03/19	27/03/19
• Préparation de la soutenance de S10	28/03/19	08/04/19

Figure 2 – Les tâches dans le diagramme de Gantt

3.Comparer et sélectionner des bibliothèques de réalité augmentée

Date de début : le 11 octobre 2018;

Date de fin : le 17 octobre 2018;

Durée : 7 jours;

4.Apprendre les bases d'ARCore

Date de début : le 18 octobre 2018;

Date de fin : le 31 octobre 2018;

Durée : 14 jours;

5.Analyse les performances d'ARCore

Date de début : le 01 novembre 2018;

Date de fin : le 15 novembre 2018;

Durée : 14 jours;

6.Test les performances d'ARCore et de la base de image

Date de début : le 16 novembre 2018;

Date de fin : le 22 novembre 2018;

Durée : 7 jours;

7.Intégration et rédaction du rapport de S9

Date de début : le 23 novembre 2018;

Date de fin : le 06 décembre 2018;

Durée : 14 jours;

8.Préparation de la soutenance de S9

Date de début : le 07 décembre 2018;

Date de fin : le 13 décembre 2018;

Durée : 7 jours;

Le semestre10 17/12/2018-05/04/2019 9.Test ARCore sur votre appareil mobile

Date de début : le 17 décembre 2018;

Date de fin : le 02 janvier 2019;

Durée : 14 jours;

10.Conception de l'interface et modélisation logicielle

Date de début : le 03 janvier 2019;

Date de fin : le 16 janvier 2019;

Durée : 14 jours;

11.Développement et programmation

Date de début : le 17 janvier 2019;

Date de fin : le 21 mars 2019;

Durée : 63 jours;

12.Test et amélioration de performance

Date de début : le 28 février 2019;

Date de fin : le 21 mars 2019;

Durée : 21 jours;

13.Rédaction du rapport S10

Date de début : le 14 mars 2019;

Date de fin : le 21 mars 2019;

Durée : 7 jours;

14.Préparation de la soutenance de S10

Date de début : le 22 mars 2019;

Date de fin : le 28 avril 2019;

Durée : 7 jours;

B

Technologies utilisées

Afin de mener à bien ce PRD je vais utiliser un certain nombre de technologies que je vais lister et décrire ici.

1 Outils pour la gestion de projet

- Un gestionnaire de version : GitKraken . C'est le logiciel de gestion de version. Il permet un versionning hors-ligne et collaboratif.
- GitHub (<https://github.com/zhangbolun2356/PRD>). GitHub permet de faciliter le développement collaboratif notamment en proposant une gestion des dépôts Git (branches, droit d'accès ...) ou une création de wiki.

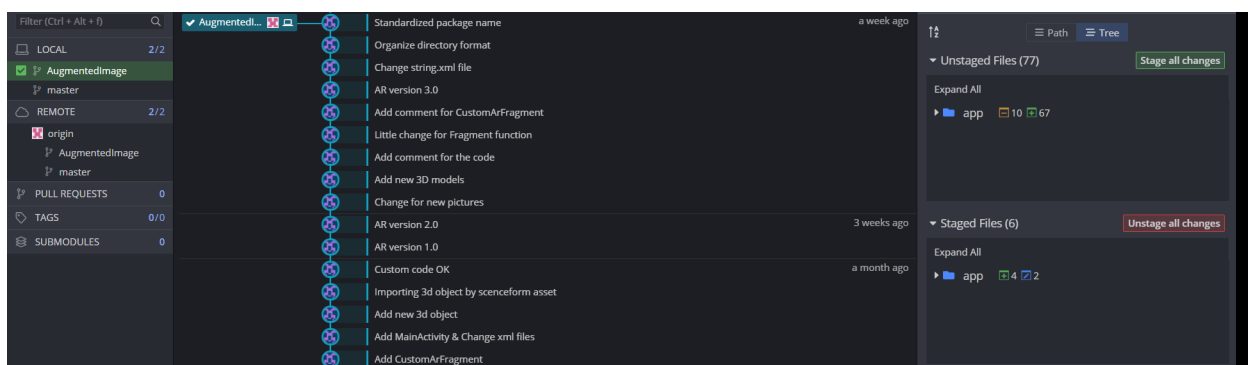


Figure 1 – Gitkraken

2 Technologies pour le développement

- Un langage de programmation : Java. Le langage Java est celui utilisé pour développer des application Android.
- Un système d'exploitation : Android. C'est le système d'exploitation mobile le plus utilisé dans le monde.

- Un IDE : AndroidStudio(<https://developer.android.com/studio>), C'est un IDE basé sur IntelliJ développé pour le développement d'applications Android.
- Un kit de développement : Android SDK27 (software development kit). C'est un ensemble d'outils pour le développement Android. Il comporte notamment un débogueur, des bibliothèques et un émulateur.
- Framework : Android Application Framework

C

Spécifications fonctionnelles

L'application implémentera la réalité augmentée à l'aide de l'outil ARCore, ainsi que d'Android SDK Platform 7.0 (API de niveau 24) ou supérieure. L'application fonctionne comme suit :

- L'utilisateur ouvre l'application mobile. L'application mobile utilisera automatiquement la caméra de l'appareil mobile et activera la fonction caméra (nécessite un accès préalable à la caméra de l'application mobile).
- L'utilisateur aligne la caméra du périphérique mobile sur l'objet 3D à identifier, l'application mobile appelle ARCore et utilise la fonction ARCore pour rechercher des images de caractéristiques correspondantes dans la base de données d'images.
- En fonction des résultats renvoyés par ARCore, le programme reconnaît avec succès l'objet 3D et implémente la réalité augmentée. L'utilisateur peut interagir avec l'objet 3D en touchant l'objet capturé par la caméra. L'interaction comprend les méthodes suivantes :
 - Afficher l'introduction de l'objet
 - Afficher les détails de l'objet
 - Afficher la vidéo d'introduction de l'objet
 - Afficher la photo correspondante de l'objet
 - Afficher les objets 3D générés par la réalité augmentée autour de l'objet et utiliser ces objets pour interagir avec l'utilisateur

Les différents affichages seront sélectionnés via un menu qui sera affiché autour de l'objet une fois que le programme aura reconnu l'objet 3D et réussi à afficher des informations différentes en cliquant sur différents modes d'affichage.

Les objets 3D générés par la réalité augmentée se déplaceront autour des objets identifiés et interagiront avec l'utilisateur (vous pouvez apprendre des informations sur les objets identifiés à travers ces objets)

D

Spécifications nonfonctionnelles

1 Contraintes de développement et conception

- langage de développement : JAVA + Android
- Système de développement : Windows
- Appareil de développement : Sumsung Galaxy Note9

2 Contraintes de fonctionnement et d'exploitation

2.1 Performances

Lors de l'utilisation de cette application, la durée de reconnaissance d'un objet 3D ne doit pas dépasser 10 secondes.

Pour les applications mobiles, le temps de reconnaissance dépend de la richesse de l'image augmentée dans la base de données d'images augmentée. Lorsque le nombre de images augmentée d'un objet 3D reconnu est important et que la différence est grande, le programme sera plus facile à identifier et, par conséquent, le temps de reconnaissance sera également plus court.

2.2 Capacités

Pour la programme, nous considérerons que la base de données d'images augmenté a une capacité de 1000 images (en raison des limitations d'ARCore), et pour chaque objet 3D reconnaissable, nous stockons 20 image augmentée (angles différents, Intensité lumineuse différents, distances différents ...)

Le nombre d'objets 3D reconnaissables par le programme sera défini entre 5 et 10, et dans les futures versions, il sera peut-être possible d'introduire plus de bases de données d'images, de sorte que le nombre d'objets 3D pouvant être reconnu sera encore augmenté.

2.3 Modes de fonctionnement

Actuellement, l'application ne prend en charge que les appareils mobiles du système Android. Les appareils mobiles pris en charge seront basés sur le fait que ARCore est pris en charge ou non. Par conséquent, les appareils mobiles pris en charge seront les mêmes que les appareils mobiles pris en charge par ARCore.

Deep learning et réalité augmentée :

BOLUN ZHANG

Encadrement : GILLES VENTURINI

Objectifs

Les principaux objectifs du projet sont:

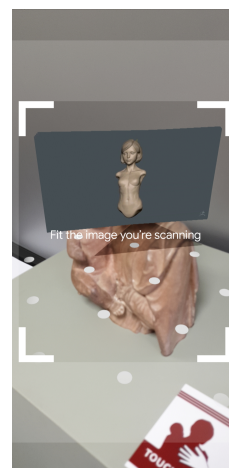
- Reconnaître des objets 3D dans le monde réel
- Implémenter l'interaction de réalité augmentée
- Prise en charge multi-appareils du système Android



Logo d'ARCore

Mise en œuvre

Le programme peut actuellement utiliser ARCore pour reconnaître **plusieurs objets 3D**, et superposant **des vidéos, des modèles 3D et des images** comme l'interaction. Les résultats de la reconnaissance et de l'interaction seront affichés sur l'écran du appareil mobile de l'utilisateur.



Reconnaissance et interaction

Résultats attendus

- Reconnaître avec des **objets 3D**
- Utiliser la réalité augmentée pour des objets reconnus
- Les utilisateurs peuvent **interagir** avec des objets reconnus
- Le programme peut identifier *plusieurs* objets 3D différents
- Peut être importés les images augmentée sur le réseau



Deep learning et réalité augmentée :

BOLUN ZHANG

Encadrement : GILLES VENTURINI

Objectifs

Les principaux objectifs du projet sont:

- Reconnaître des objets 3D dans le monde réel
- Implémenter l'interaction de réalité augmentée
- Prise en charge multi-appareils du système Android

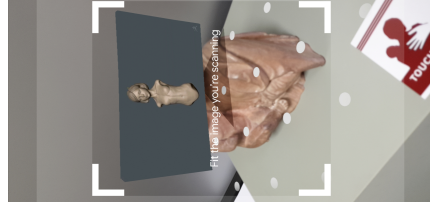
Mise en œuvre

Le programme peut actuellement utiliser ARCore pour reconnaître **plusieurs objets 3D**, et superposant **des vidéos, des modèles 3D et des images** comme l'interaction. Les résultats de la reconnaissance et de l'interaction seront affichés sur l'écran du appareil mobile de l'utilisateur.



ARCore

Logo d'ARCore



Reconnaissance et interaction



En collaboration avec Polytech Tours

Résultats attendus

- Reconnaître avec des **objets 3D**
- Utiliser la réalité augmentée pour des objets reconnus
- Les utilisateurs peuvent **interagir** avec des objets reconnus
- Le programme peut identifier *plusieurs* objets 3D différents
- Peut être importés les images augmentées sur le réseau



Deep learning et réalité augmentée

Résumé

Le projet de recherche et développement vise à développer un logiciel qui utilise la bibliothèque de réalité augmentée ARCore dans l'environnement Android pour identifier des objets 3D dans le monde réel. Les utilisateurs découvriront les objets identifiés via le téléphone / la tablette. La reconnaissance des objets 3D est obtenue en faisant correspondre les images de caractéristiques dans la base de données d'images d'ARCore. Lorsque les objets sont bien reconnus, la bibliothèque AR utilise ces textures pour fournir un contenu numérique (images, sons, animations 3D, etc.).

Mots-clés

ARCore, Reconnaissance d'objets 3D, Android

Abstract

The Research and development project aims to develop a software that uses the ARCore augmented reality library in the Android environment to identify 3D objects in the real world, and users will discover the identified objects through the phone/tablet. The recognition of 3D objects is achieved by matching feature images in ARCore's image database. When objects are well recognized, the AR library uses these textures to deliver digital content (images, sounds, 3D animations, etc.).

Keywords

ARCore, 3D object recognition, Android

Entreprise

Polytech Tours



Tuteur entreprise

GILLES VENTURINI (Professeur d'Informatique)

Étudiant

BOLUN ZHANG (DI5)

Tuteur académique

GILLES VENTURINI