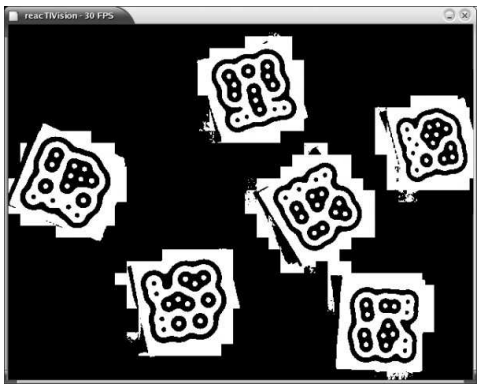
Multimodal libraries
ReacTiVision



Auteurs : **Luca Zingg, Daniele Della Bruna et Yannick Thiessoz**
Superviseur : **Denis Lalanne, Omar Abou Khaled et Jacques Bapst**
Classe : **Master en Informatique université de Fribourg**
Type : **Projet de phase I du cours MultiModal Interfaces**
Début : **13.03.2006**
Fin : **08.05.2006**

1 Introduction

La première partie du projet du cours Multimodal interfaces permet de faire un survol des librairies multimodales de l'état de l'art. Ces librairies sont en quelque sorte les boîtes à outils permettant la réalisation de projets multimodaux. Elles recouvrent les modalités suivantes : voix, gestes, expressions du visage, visualisation, sonification etc. Chaque groupe d'étudiants se concentrera sur une librairie et la présentera aux autres étudiants.

Notre groupe a eu l'opportunité de tester la librairie multimodal **ReacTiVision**. Ce rapport va vous permettre de vous familiariser avec cette librairie. Il vous permettra de connaître les buts de cette librairie, son fonctionnement et ses limites. Ce document vous permettra aussi de voir en détail la capacité de cette librairie à être intégrée au sein d'un petit projet.

2 La librairie ReacTiVision

ReacTiVision est plus qu'une simple librairie, reacTiVision est un framework permettant le développement rapide d'applications. ReacTiVision est aussi open source et multiplateforme ce qui simplifie et augmente ses possibilités d'utilisation.

Sa fonction principale est le "tracking" de marqueur. Cela signifie que ce framework peut suivre le cheminement de certains marqueurs de manière robuste. Il a été développé pour permettre la réalisation rapide d'une interface utilisateur tangible basée pour une utilisation sur une table.

Son développement a été réalisé dans le cadre du projet reacTable (**multi-user electro-acoustic music instrument with a tabletop tangible user interface**) par Monsieur Martin Kaltenbrunner. Son moteur de suivi des marqueurs utilise la librairie fidtrack de Ross Bencina qui est une implémentation plus performante du concept d-touch d'Enrico Costanza.

2.1 Fonctionnement de la librairie

ReacTiVision est un framework divisé en deux parties. La première partie étant l'application reacTiVision qui transmet des messages « OpenSound Control » sur un réseau UDP à la seconde partie qui est une application client (backend).

L'application reacTiVision implémente le protocole TUIO qui permet de transmettre l'état d'objet tangible se trouvant sur une table.

Son fonctionnement est donc le suivant :

Partie 1	1. Suivi des marqueurs à l'aide d'une caméra 1.1. Transformation du stream video en noir et blanc 1.2. Découpage de l'image en arbre de région b/n 1.3. Recherche dans cet arbre des marqueurs 1.4. Recherche de la correspondance entre ce marqueur et son ID dans un dictionnaire 1.5. Insertion des informations (présence, position, orientation, identité...) dans le message « OpenSound Control » qui implémente le protocole TUIO
	2. Transmission des messages « OpenSound Control » via le protocole TUIO vers l'application cliente.
Parite 2	3. Récupération des ces messages par l'application cliente(backend) 4. Traitement des informations

Les marqueurs sont des images spécialement étudiées pour le tracking.

En voici quelques exemples :

2.2 Possibilités offertes par la librairie

Voici la liste des possibilités qu'elle offre :

- Tracking de marqueurs :
  - Position
  - Angle
  - Déplacement speed
  - Rotation speed
  - Déplacement accélération
  - Rotation accélération
  - ClassID (type de marqueur)
  - SessionID(plusieurs marqueurs du même type en simultané)

- L'application reacTiVision est disponible pour les plateformes suivantes ainsi qu'en code source à compiler pour les autres plateformes :
  - Win32
  - MacOs X
  - Linux

- Disponibilité du framework de développement de la partie client pour les langages suivants:
  - C++
  - Java
  - Processing
  - Pure Data
  - Flash

- Envoi des données trackées par l'application reacTiVision vers l'application cliente au travers d'un réseau physique.

Les possibilités sont malgré tout limitées à la modalité du tracking optique de marqueurs. Il n'y pas d'autres modalités tel que le son qui ne soit intégrée à ce framework, tout comme il n'y a pas d'autre moyen de tracking comme le magnétisme.

De plus, comme nous vous le spécifierons dans la partie de tests, le fonctionnement du tracking est limité à l'environnement (qualité de la camera, luminosité de la pièce, ...).

Il n'est pas possible non plus d'utiliser ces propres marqueurs. Par contre, un grand nombre de marqueurs prédéfinis sont disponibles (90).

2.3 Utilisation et installation

L'utilisation ainsi que l'installation sont très simples. Dans un premier temps, il est possible d'utiliser le simulateur (TUIO_Simulator.zip) qui est fourni sur le site de la librairie. Grâce à ce simulateur vous n'avez pas besoin de table, de camera ni même de marqueurs. En effet, cette application java simule une table sur laquelle on peut placer des marqueurs virtuels et elle envoie leurs mouvements vers la partie backend (par défaut localhost sur le port 3333). Cela peut être intéressant lors du développement d'un projet (pas besoin d'un grand nombre d'équipements).

Dès le moment où l'on ne veut plus utiliser le simulateur mais les composants réels, les étapes suivantes sont nécessaires :

- Impression des marqueurs (fiducials.pdf)
- Téléchargement de l'application reacTiVision
    - Disponible pour les plateforme (Win32, MacOS X, Fedora 4, Linux)
    - Pour les autres plateformes il faut encore compiler le source.
- Brancher la caméra
- Si nécessaire calibrer l'application reacTiVision, seulement dans le cas d'utilisation de miroirs que nous n'allons pas traiter.
- Lancer l'application reacTiVision
    - Options de démarrage
        - Backend ipAddress portNumber (default localhost 3333)
        - Ne pas analyser l'image complète mais qu'une partie (ex : table ronde)
        - Choix du set de marqueurs (amoeba, classic, d-touch)
    - Options après le lancement
        - Affichage (image source, noir blanc et sans image)
        - Sauvegarde d'image (bmp ou raw)
        - Affichage du numéro des images reconnues sur l'ecran
- Lancer l'application backend

La partie sans doute la plus intéressante est la partie backend. Tout d'abord la partie backend la plus simple est celle fournie sur le site. Elle permet (version java et c++) soit d'afficher les messages directement sur la console soit de voir les objets se déplacer sur une table virtuelle. Pour l'utilisation de backend plus avancé voir le chapitre suivant.

2.4 Intégration

Pour l'intégration de la librairie reacTiVision au sein d'un projet plus large il y a deux possibilités. La première est la modification de l'application reacTiVision qui s'occupe du suivi des marqueurs et du transfert des messages vers la partie backend. Cela est rendu possible par le fait que reacTiVision est open source. Mais ce n'est pas l'intégration la plus fréquente de cette librairie.

La manière commune d'intégrer cette librairie au sein d'un projet est en réaliser la partie backend. Pour réaliser une application cliente il faut les deux fichiers (qui sont fournis avec les applications de démonstrations) suivants :

- **TuioClient** : Il représente la classe qui va écouter les « OSCMessage » qui sont envoyés par l'application reacTiVision et fournir ces messages aux listeners qui se sont enregistrés auprès de cette classe.

- **TuioListener :** Cette classe est l'interface que les listeners qui veulent recevoir les messages du TuioClient doivent implémenter.

  Les méthodes définies dans l'interface sont les suivantes:

  - **addTuioObj(long s_id, int f_id):**

    Appelée dès qu'un objet arrive sur la table

  - **removeTuioObj(long s_id, int f_id):**

    Appelée dès qu'un objet n'est plus sur la table

  - **updateTuioObj(long s_id,**
    int f_id,
    …):

    Appelée dès q'un objet change d'état et que ces données (position, direction...) doivent être mises à jour.

  - **refresh():**

    Appelée à chaque envoie d'une image depuis reacTiVision même s'il n'y a aucune update d'objet. Permet par exemple de mettre à jour la vue.

Mise à part ces deux fichiers, il faut aussi la bibliothèque de fichiers pour les messages de type « OpenSound Control ». Ces fichiers sont aussi disponibles dans les exemples.

Avec ces fichiers vous voilà parés pour réaliser une application qui tire partie des possibilités offertes par la librairie.

2.5 Tests

Nous avons réalisé quelques tests sur la librairie et la caméra. Ces tests sont de trois types :

- Tracking de la cible en fonction de sa taille et sa distance avec la caméra.
- Tracking de la cible avec la modification de l'angle.
- Tracking de la cible au travers une vitre opaque.

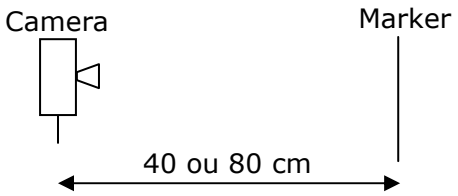
Les tests ont étés effectués à l'aide d'une Webcam ayant les caractéristiques suivantes :

- Frame per second : 15
- Color space / compression : UYVY
- Output size : 640*480 or 320*240

2.5.1 Tracking distance

Cette mesure nous permet d'éprouver le tracking de la cible en fonction de la distance de cette cible, de sa taille et de la résolution de la sortie. Pour que ce suivi de cible soit considéré comme valide il faut pouvoir bouger la cible pendant 10 secondes sans que le session_id ne change.

Voici un graphique ainsi que les résultats de tests :

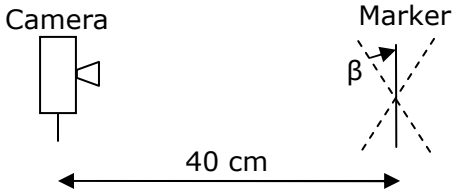


	Cible à 40 cm		Cible à 80 cm	
Taille de la cible	Capteur 640*480	Capteur 320*240	Capteur 640*480	Capteur 320*240
5.5cm	Y	Y	Y	Y
5cm	Y	Y	N	Y
4.5cm	Y	Y	N	N
3.5cm	Y	Y	N	N
3cm	Y	N	N	N
2.5cm	N	N	N	N

2.5.2 Tracking angle

Cette mesure nous permet d'éprouver le tracking de cible de taille donnée (5 cm) à une distance donnée (40 cm) en fonction d'un angle de prise de vue et de la résolution de la sortie. Pour que ce suivi de cible soit considéré comme valide il faut pouvoir bouger la cible pendant 10 secondes sans que le session_id ne change.

Voici un graphique ainsi que les résultats de tests :

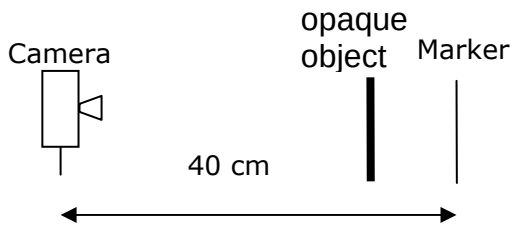


Angle β	Capteur 640*480	Capteur 320*240
60°	Y	Y
65°	Y	N
70°	Y	N
75°	N	N

2.5.3 Tracking opacity

Cette mesure nous permet de d'éprouver le tracking de cible à une distance donnée (40 cm) avec un angle de prise de vue de 0° en fonction de la taille de la cible et de la résolution de la sortie. Cette mesure a été réalisée à l'aide d'un objet opaque. Elle est utile pour montrer qu'il serait possible d'avoir une table en verre opaque à travers lequel l'on pourrait faire la reconnaissance de cible dans le but d'une éventuelle projection de données sur ce verre. Pour que ce suivi de cible soit considéré comme valide il faut pouvoir bouger la cible pendant 10 secondes sans que le session_id ne change.

Voici un graphique ainsi que les résultats de tests :



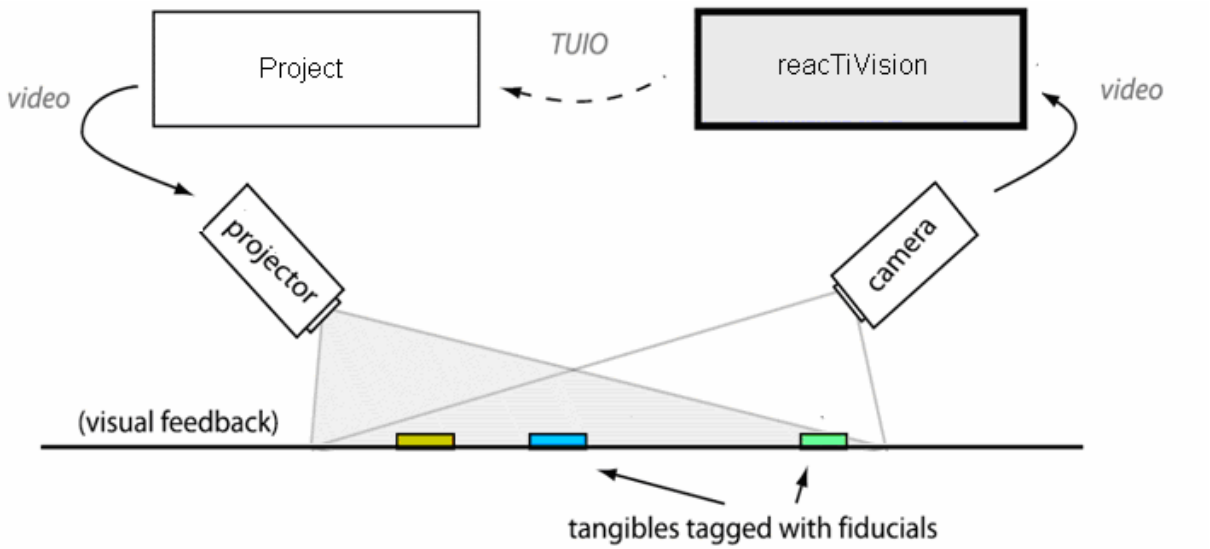
Taille de la cible	Capteur 640*480	Capteur 320*240
4cm	Y	Y
3.5cm	Y	N
3cm	N	N

3 Notre démonstration de la librairie

Notre projet consiste à utiliser la librairie reacTiVision pour dessiner sur une table de la même manière qu'un stylo sur une feuille. Il doit permettre d'utiliser plusieurs couleurs et de pouvoir effacer les erreurs de dessin.

3.1 Fonctionnement

Le fonctionnement s'apparente donc à un stylo sur une feuille. Où la feuille serait la table, les stylos seraient les cibles et la marque laissée pas les cibles serait émulée par le projecteur comme le montre le schéma suivant.



Pour réaliser cela nous avons attribué une couleur à chaque cible. Le placement de la cible sur la table équivaut au positionnement du stylo sur la feuille. De là, le déplacement du stylo produit un trait continu sur la table.

Pour effacer nous avons un stylo sans couleur. Pour déterminer quand la cible doit commencer nous avons mis en place, grâce à la détection de la rotation de la cible, un système qui permet de ne pas écrire dans un quart de la rotation complète et d'écrire dans les autres trois quarts.

3.2 Installation

L'installation de notre projet peut être découpée en deux phases.

3.2.1 La mise en place matériel

Cette partie comprend :

- Branchement du projeteur et de la caméra à l'ordinateur qui contient notre projet et la librairie.
- La mise en place de la caméra et réglage du focus.
- La mise en place de la table et du projecteur pour que l'image projetée soit de la bonne dimension et au bon endroit sur la table par rapport a ce que détecte la caméra.
- Choix de la taille des cibles en fonction de la détection de celle-ci par la caméra.

3.2.2 La mise en place logiciel

Cette partie comprend :

- Lancement de la librairie avec une résolution de la camera de 640 x 480 et toutes les autres options laissées comme default.
- Lancement de notre application java (java –jar DemoYaDaLu.jar).

3.3 Tests

Pour tester notre application nous avons réalisé un dessin représentant un paysage simple. Ce test est disponible sur le cd en annexe sous forme d'une petite vidéo.

Nous ne sommes pas très satisfaits de notre démo car avec les tests que nous avons effectués sans l'aide de la projection par beamer la cible était presque toujours détectée, tandis qu'en projetant l'image où nous allons filmer les lignes déjà dessinées nous empêchent la reconnaissance des cibles.

4 Conclusions

Notre petite application permet de voir les facettes les plus intéressantes de la librairie. A savoir le suivi robuste de cibles en conditions optimales aussi que la possibilité de récupération de l'orientation d'une cible rendue possible par leur design. Nous n'avons pas mis en avant certaines possibilités de la librairie tel que le transfert des informations capturées pas l'application reacTiVision à travers le réseau. De plus, nous n'avons pas jugé utile de mettre en avant des informations pouvant être déduites d'une suite de déplacement comme la vitesse de la cible ou son accélération. Par contre, le projet nous permet d'apprécier les capacités d'une telle librairie.

5 Références

Le site de la librairie :

http://sourceforge.net/projects/reactivision/

http://www.iua.upf.es/mtg/reactable/?software

Site du projet reacTable :

http://www.iua.upf.es/mtg/reacTable/

Site de Monsieur Ross Bencina créateur de la librairie fidtrack :

http://www.audiomulch.com/~rossb/

Site de Monsieur Enrico Costanza créateur du framework d-touch :

http://web.media.mit.edu/~enrico/

Site sur les messages de type OpenSound Control :

http://www.cnmat.berkeley.edu/OpenSoundControl/

PDF sur le protocole TUIO :

http://www.iua.upf.es/mtg/reacTable/pdfs/GW2005-KaltenBoverBencinaConstanza.pdf

PDF sur les marqueurs :

http://www.iua.upf.es/mtg/reacTable/pdfs/reactivision_3rditeration2005.pdf

PDF sur la recherche des marqueurs :

http://www.iua.upf.es/mtg/reacTable/pdfs/procams05-BencinaKaltenJorda.pdf

6 Annexes

Vous trouverez sur le cd les documents suivants :

- Toutes les sources et les différentes .pdf disponibles sur le site de la librairie.
- Le projet Eclipse de la démo.
- Un jar exécutable de la démo.
- La vidéo de la démo.
- Le rapport.
- La présentation .ppt