

Projet Encadré "Image Vision Interaction": Surface Multi-touch

François Cabestaing, Géry Casiez, Olivier Losson, Fabrice Aubert

▶ **To cite this version:**

François Cabestaing, Géry Casiez, Olivier Losson, Fabrice Aubert. Projet Encadré "Image Vision Interaction": Surface Multi-touch. Colloque CETSIS, Oct 2011, Québec, Canada. inria-00638407

HAL Id: inria-00638407
https://inria.hal.science/inria-00638407

Submitted on 3 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Projet Encadré « Image Vision Interaction » : Surface Multi-touch

François Cabestaing, Géry Casiez, Olivier Losson et Fabrice Aubert
Université Lille 1, Sciences et Technologies
Cité Scientifique, 59655 Villeneuve d'Ascq Cedex, FRANCE

***Résumé*— Dans cet article, nous décrivons le contenu pédagogique d'une unité d'enseignement intitulée « projet encadré Image, Vision, Interaction ». Son objectif est de présenter, en seulement douze séances de quatre heures, la plupart des notions d'analyse et de synthèse d'images à des étudiants de première année de master. Le fil conducteur de ce projet encadré est l'étude du fonctionnement et de l'exploitation d'une surface multi-touch intégrant une caméra.**

***Mots clés*— surface multi-touch, vision, image, interaction, pédagogie par projet.**

I. Introduction

DANS le domaine des sciences pour l'ingénieur, l'image a depuis longtemps été étudiée par deux communautés. Adoptant une approche système, les automaticiens traitent et analysent le contenu des images issues de capteurs afin d'en extraire des mesures. Pour leur part, les informaticiens modélisent la réalité pour reproduire, via des images synthétiques, une représentation réaliste d'une scène et des objets qu'elle contient. De plus en plus d'applications actuelles nécessitent d'aborder le problème selon ces deux approches. Par exemple, en imagerie per-opératoire, les données du patient mesurées en temps-réel sur des images scanner ou IRM (aspect vision) peuvent être présentées en 3D au chirurgien de façon à simplifier leur interprétation (aspect synthèse).

Dans la spécialité « Image, Vision, Interaction » (IVI), notre objectif est de former les étudiants à toutes ces facettes des métiers de l'image. Cette spécialité est adossée à deux mentions du master « Sciences, Technologies, Santé » de l'Université Lille 1, à savoir la mention informatique et la mention automatique et systèmes électriques (ASE). Durant la première année de Master, quatre modules spécifiques sont suivis en commun par les étudiants des deux mentions qui veulent intégrer la spécialité IVI. L'un de ces modules est enseigné au premier semestre, donc avant que les étudiants n'aient vraiment réalisé un choix parmi les spécialités offertes en deuxième année.

L'objectif de ce module, enseigné sous la forme d'un projet encadré, est de concrétiser une bonne partie des notions d'analyse et de synthèse d'images afin que les étudiants puissent faire un choix motivé de leur spécialité. Le projet encadré peut être considéré comme un super-TP qui se déroule sur une cinquantaine d'heures. Il offre la possibilité aux étudiants de vérifier si les notions qui seront abordées par la suite de façon beaucoup plus formelle les intéressent suffisamment pour qu'ils choisissent cette spécialité en M2.

Dans cet article, nous décrivons en premier lieu le support matériel qui permet de réaliser la plupart des manipulations lors des séances de projet, à savoir une surface multi-touch conçue et fabriquée par une partie de l'équipe pédagogique du master. Dans la deuxième partie de l'article, nous présentons le contenu des différentes séances du projet encadré IVI.

François Cabestaing (francois.cabestaing@univ-lille1.fr) et Géry Casiez (gery.casiez@lifl.fr) sont co-responsables de la spécialité de Master « Image, Vision, Interaction » de l'Université Lille 1, Sciences et Technologies.

II. Maquette de TP

Chaque binôme d'étudiants dispose d'une maquette pour réaliser les manipulations prévues dans les séances de travaux pratiques. La maquette est constituée d'un boîtier en PVC de 8 mm d'épaisseur, au fond duquel est placée une caméra. Cette dernière observe la face supérieure du boîtier, constituée d'une épaisse plaque de plexiglas, comme le montre la photographie de la figure 1.

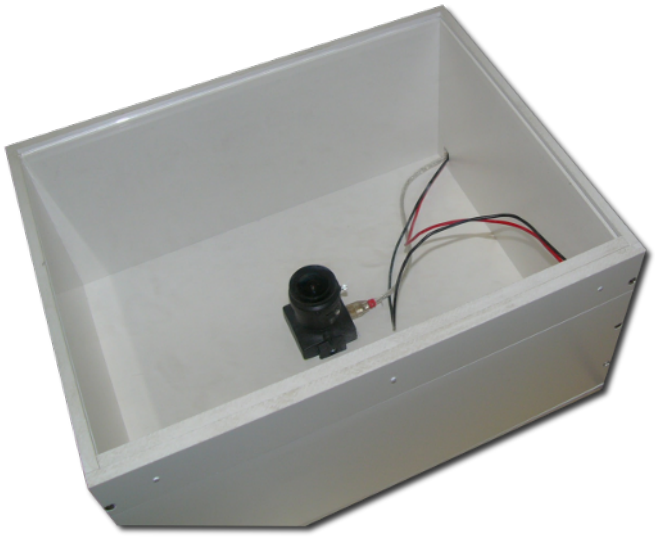


Fig. 1. Boîtier utilisé pour les travaux pratiques

Les positions des « curseurs » qui permettent l'interaction avec l'application sont déterminées en mettant en évidence les doigts de l'utilisateur dans les images acquises par la caméra. C'est plus précisément le contact entre l'extrémité du doigt de l'utilisateur et la plaque de plexiglas qui est détecté. Dans notre maquette, c'est la technique FTIR (*Frustrated Total Internal Reflection*) qui est utilisée pour faire clairement apparaître les zones de contact dans les images [1].

Pour éclairer l'intérieur de la plaque de plexiglas, une série de cinq LEDs infrarouge est installée sur une partie détachable de l'un des côtés du boîtier. Cette partie est représentée sur la photographie de la figure 2. Les LEDs sélectionnées (OSRAM SFH 484-2) ont un demi-angle d'émission de 8°, suffisamment étroit pour que les pertes de puissance lumineuse lors des réflexions des rayons sur les faces du plexiglas soient minimales. Leur longueur d'onde de 880 nm a été retenue afin que les perturbations introduites par les autres sources de lumière présentes dans la salle soient faibles.



Fig. 2. Diodes électroluminescentes infrarouge utilisées pour éclairer la plaque de plexiglas

Nous avons sélectionné la caméra USB UI-1225LE-C du constructeur IDS [1] pour acquérir les images. À cette caméra est associé un objectif de très courte longueur focale (TAMRON zoom 2,2 mm corrigé IR), de telle sorte que le champ d'observation couvre la totalité de la face supérieure du boîtier. Contrairement aux solutions couramment présentées pour constituer une surface multi-touch, qui reposent souvent sur l'utilisation d'une webcam, l'utilisation d'une caméra de gamme semi-professionnelle présente plusieurs avantages.

En premier lieu, le capteur utilisé sur la caméra choisie intègre une matrice de filtres couleur (CFA, *Color Filter Array*) de type Bayer. Chaque photosite est ainsi recouvert d'un filtre rouge, vert ou bleu, ce qui permet de disposer d'images couleur après dématriçage [4]. Cependant, la réponse spectrale des trois types de filtres CFA devient quasiment identique dans l'infrarouge[1], ce qui permet d'acquérir des images dans cette gamme de longueurs d'onde sans être perturbé par la matrice de filtres. Cette caméra permet donc d'acquérir soit des images infrarouge en plaçant devant le capteur un filtre passe-bande à 880 nm, soit des images couleur grâce à un filtre neutre coupant l'infrarouge.

D'autre part, contrairement à la plupart des webcams, la caméra IDS est commercialisée avec un pilote logiciel et une bibliothèque de fonctions qui autorise le réglage de tous ses paramètres de fonctionnement. Cela permet d'étudier toutes les notions liées à l'acquisition d'images numériques durant les séances de travaux pratiques : horloge pixel, cadence d'acquisition, binning, temps d'intégration, gains, taille et position d'une fenêtre d'intérêt, etc.

Lorsque les images sont acquises en mode « infrarouge », les zones correspondant aux contacts entre les doigts et la plaque de plexiglas apparaissent clairement, comme on peut le constater sur les images de la figure 3. Du fait de la très courte longueur focale de l'objectif, les images sont très distordues. Ce phénomène peut être corrigé sans problème en réechantillonnant les images après acquisition.

1. IDS Imaging, Documentation technique de la caméra UI-1225LE-C

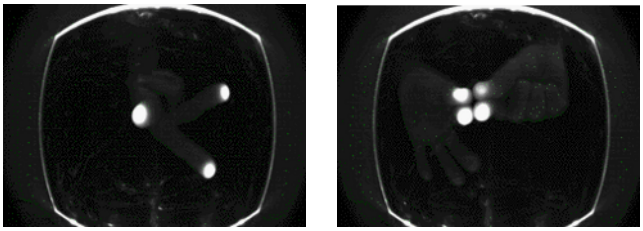


Fig. 3. Exemples d'images obtenues en mode « infrarouge » grâce au procédé FTIR.

III. Déroulement du projet encadré

Le projet encadré « surface multi-touch » se déroule durant douze séances de quatre heures. Chaque séance comporte une à deux heures de cours traitant d'un des éléments, matériel ou logiciel, de la « chaîne » qui va de la formation des images FTIR jusqu'au contrôle d'une application graphique par une surface multi-touch. Chaque cours est complété par une séance de travaux pratiques, durant laquelle les étudiants expérimentent de façon concrète les concepts qui ont été abordés durant le cours.

Naturellement, comme les notions étudiées durant les douze séances sont à la fois très nombreuses et très variées, les travaux pratiques sont assez différents de ceux qu'on enseigne habituellement dans des unités plus disciplinaires. Dans certains cas, les étudiants doivent simplement ajuster les paramètres d'un programme déjà développé, en vue de comprendre leur influence sur le fonctionnement du système. Dans d'autres cas, il leur faut compléter un code déjà opérationnel fourni en début de séance, afin d'implanter une ou deux fonctions bien précises. Tous les énoncés et codes supports sont évidemment disponibles en ligne [2], ce qui permet aussi de tirer parti de l'applet OpticsApplet [3] pour les expérimentations sur la partie optique.

A. Aspects « bas niveau »

Cette première partie du projet regroupe six séances et concerne les aspects qu'on peut qualifier de « bas niveau » du système multi-touch : principes de l'optique géométrique, phénomène FTIR, propriétés et réglage des paramètres d'une caméra numérique, acquisition des séquences d'images numériques, traitements ponctuels et/ou locaux d'une image, analyse en composantes connexes, mise en correspondance pour le suivi des composantes connexes. Cette partie des enseignements est assurée par les « automaticiens » de l'équipe pédagogique.

Parmi les étudiants concernés, certains ont peu d'expérience pratique de la programmation. Cela a conduit au choix de la bibliothèque OpenCV [4] pour réaliser simplement toutes les opérations nécessaires à la détection en temps réel des doigts dans les images. Implémentée en C, OpenCV offre en effet des fonctions préprogrammées pour la plupart des traitements bas niveau classiques : acquisition (par lecture du buffer de la caméra IDS) et affichage des images, calcul

2. http://master-ivi.univ-lille1.fr/Cours/PjE
3. http://webphysics.davidson.edu/applets/Optics4
4. http://opencv.willowgarage.com

de l'histogramme, binarisation et segmentation, filtrage, etc. Toujours dans un souci de simplicité du développement, nous utilisons une version non-objet de la bibliothèque, bien que la version utilisée (2.0) inclut une interface C++.

À titre d'exemple, la dernière séance de cette partie concerne le suivi des empreintes des doigts sur la plaque de plexiglas. La séance précédente a permis de détecter l'ensemble des pixels (*blobs*) correspondant à chaque doigt grâce à un étiquetage en composantes connexes. Le suivi consiste alors à associer sans ambiguïté la même étiquette à un même doigt tout au long du déplacement de celui-ci. Autrement dit, pour chaque image de la séquence, chaque blob doit être assigné à l'un des doigts (alors déterminé comme un « objet » singulier), de sorte que chaque doigt conserve toujours la même étiquette (voir figure 4).

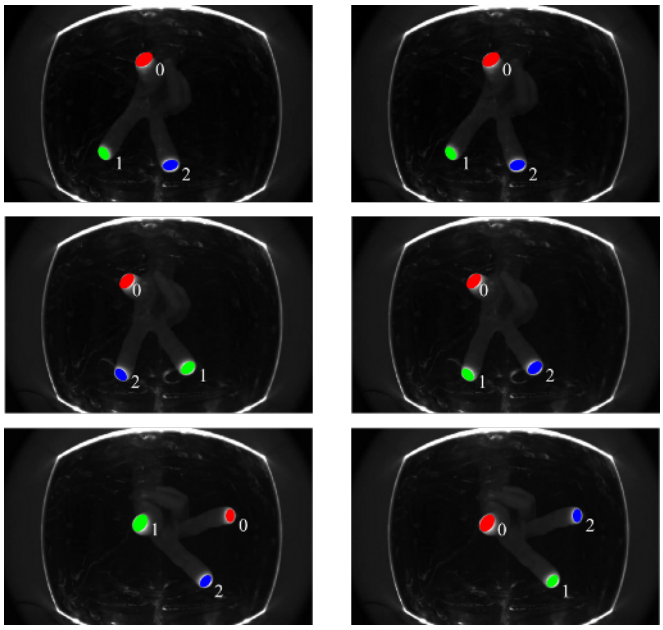


Fig. 4. Nécessité du suivi, sur un exemple de mouvement de rotation anti-horaire de trois doigts. Les images n°0, 10 et 40 de la séquence sont présentées verticalement. Colonne de gauche : sans suivi, les blobs sont étiquetés dans l'ordre où ils sont rencontrés dans le parcours classique des pixels. Colonne de droite : avec un suivi effectif, les blobs conservent leur étiquette durant le mouvement de rotation.

Pour cela, les étudiants doivent compléter la trame de programme fournie pour implémenter un algorithme de mise en correspondance simple, qui consiste à assigner à chaque blob de l'image courante l'étiquette du blob qui est le plus proche dans l'image précédente. Les étudiants les plus avancés sont invités à implémenter un second algorithme basé sur une distance inter-blobs généralisée. Un second exercice du TP vise à résoudre les problèmes d'apparition et de disparition de blobs dans une image par rapport à l'image précédente.

À l'issue de cette première série de séances, les résultats du suivi des doigts doivent être enfin transmis à l'étage de « haut niveau » du système multi-touch, ce qui constitue l'exercice final du dernier TP. Pour cela, nous mettons en œuvre le protocole TUIO[5], précisément conçu pour les interfaces tactiles [3]. La partie « bas niveau » de l'application multi-touch implémente un serveur TUIO, grâce auquel elle transmet ses résultats à la partie « haut niveau » qui joue le rôle de client TUIO. Les messages TUIO contiennent les informations relatives aux objets suivis, qui peuvent être plus ou moins complexes (identifiant, position, vitesse et accélération, vitesse et accélération angulaires, taille et volume, etc.) selon les besoins de l'application (surface 2D, 2.5D, 3D). Un simple « curseur2D » suffit dans notre cas, puisqu'il contient l'identifiant du doigt, ainsi que sa position et sa vitesse dans le plan. Des applications développées par le groupe NUI[6], telles que Community Core Vision (CCV) ou TuioDump, permettent de contrôler l'envoi et le contenu des messages par le serveur TUIO (voir figure 5).

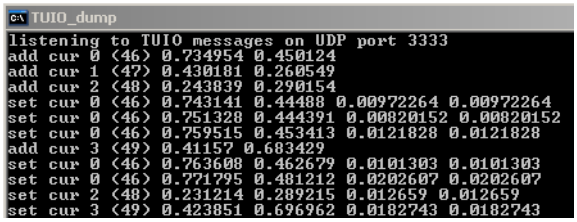


Fig. 5. Exemples de messages TUIO visualisés sous TuioDump : trois doigts sont d'abord détectés, dont les « curseurs » contiennent l'identifiant et la position (x, y). Le curseur n°0 bouge ensuite par deux fois aux positions et aux vitesses instantanées $(\delta x/\delta t, \delta y/\delta t)$ indiquées. Puis apparaît un quatrième curseur, etc.

B. *Aspects « haut niveau »*

Un deuxième groupe de six séances concerne les aspects « haut niveau » du système multi-touch. L'objectif est de développer les bases d'une boîte à outils permettant d'utiliser ces différents points de contact pour interagir avec des objets graphiques. L'application permettant de tester cette boîte à outils est une manipulation de photos comme illustrée sur la figure 6. Les fonctionnalités mises en oeuvre sont : la sélection (amener une photo au premier plan), le positionnement avec plusieurs doigts (mouvements de translation, rotation et zoom simultanés), l'ajout/suppression d'une photo et la réinitialisation de l'application en effectuant des gestes particuliers (croix, rectangle, cercle).



Fig. 6. Application à obtenir : les photos peuvent être manipulées en interagissant avec un ou plusieurs doigts

5. http://www.tuio.org

6. http://nuicode.com

Cette seconde partie permet d'aborder les aspects d'architecture logicielle avec l'utilisation des patrons de conception composite et observeur/observé pour gérer les composants (i.e. les objets graphiques manipulés) ainsi que les mécanismes de génération et d'abonnements aux événements (les informations provenant du serveur TUIO doivent être traduites en messages qui sont envoyés aux composants concernés). Des aspects de programmation graphique sont également abordés avec le tracé et le positionnement des composants (dont les concepts de contexte graphique, de matrice homogène et de changement de repères qui sont communs à de nombreuses librairies 2D/3D). Enfin, l'interaction avec l'application est enrichie avec une introduction à la reconnaissance de gestes de commandes (reconnaissance et déclenchement d'une commande lors du tracé d'un rectangle, d'une croix ou d'un cercle avec un doigt).

Le langage choisi pour cette partie du projet est le Java en exploitant les nombreux outils intégrés dans ce langage : la librairie d'interface *Swing* (dont l'architecture offre en outre un bon modèle pour concevoir la boite à outils multi-touch), ainsi que la librairie *Java2D* pour les tracés des composants.

Lors des travaux pratiques, les étudiants sont invités à compléter la partie d'analyse des informations provenant du serveur TUIO pour générer des événements (la connexion bas niveau au serveur TUIO ainsi que la structuration des classes de l'application sont fournies), à tracer les composants, à coder la gestion des composants (localisation des doigts par rapport aux rectangles des images et repositionnement de ces images en fonction du mouvement de deux doigts), puis à implémenter l'analyse d'un geste pour créer et supprimer un composant (l'algorithme présenté en cours est la méthode *$1 Recognizer* [5]).

A titre d'exemple de séance, la dernière se focalise sur la gestion des translations, rotations et changements d'échelle quand plus de deux doigts sont posés sur un objet. Pour cela les étudiants voient les notions de matrice d'inertie, calcul de valeurs propres et vecteurs propres afin de déterminer la direction principale des points en contact et suivre son évolution au cours du temps. Une seconde application du calcul de vecteurs propres est la détermination de boîtes englobantes orientées qui permet, dans l'exemple présenté aux étudiants, de faire apparaître une image à la place d'un rectangle dessiné par un doigt sur la surface et préalablement reconnu par l'application grâce au système de reconnaissance de gestes (Figure 7).

Notons enfin que l'outil de gestion de versions *Subversion*[7] est présenté et que chaque binôme dispose d'un dépôt pour développer son projet.

IV. Conclusion

Dans cet article, nous avons présenté le contenu pédagogique d'un projet encadré visant à analyser le fonctionnement d'une surface multi-touch. L'objectif de cette unité d'enseignement est de permettre à des étudiants d'étudier et de mettre en pratique des techniques d'analyse d'image et d'interaction en seulement quelques séances. Comme ce projet n'a pour l'instant été enseigné que deux fois, durant les années universitaires 2009-2010 et 2010-2011, il est impossible d'analyser de façon tangible son impact sur la réussite des étudiants. En revanche, les premiers retours de la part des étudiants sont globalement très positifs. Ils apprécient l'approche « pédagogie par projet », qui débute avec ce projet encadré et dont la part dans les enseignements progresse au cours des semestres : 5 ECTS au S1, 10 ECTS au S2 et enfin 15 ECTS au S3 pour un projet de 150 heures réalisé par des groupes de quatre étudiants.

7. http://subversion.apache.org/



Fig. 7. Capture d'écran de l'application illustrant l'utilisation du calcul de matrice d'inertie pour la détermination de boîte englobante orientée. Le rectangle rose est dessiné par l'utilisateur avec un doigt posé initialement sur l'arrière plan de l'application (en bleu). Ce rectangle est alors reconnu par l'application grâce au système de reconnaissance de gestes. L'ensemble des points du rectangle dessiné par l'utilisateur est utilisé, après rééchantillonnage, pour le calcul de la matrice d'inertie et celui des vecteurs propres (en rouge et vert). Le vecteur correspondant à la direction principale (rouge) est choisi (orange). Dans le repère orienté correspondant, les valeurs minimales et maximales en abscisse et ordonnée permettent d'obtenir les dimensions de la boîte englobante pour ensuite l'afficher (rectangle noir). Une image, redimensionnée à celle du rectangle noir, est enfin affichée à la place de ce rectangle.

References

[1] Jefferson Y. Han., "Low-cost multi-touch sensing through frustrated total internal reflection". In Proceedings of the 18th annual ACM symposium on User interface software and technology (UIST '05). 2005. ACM, New York, NY, USA, pp. 115-118, http://cs.nyu.edu/~jhan/ftirsense/.

[2] IDS Imaging, Documentation technique de la caméra UI-1225LE-C, http://www.ids-imaging.com/frontend/products.php?cam_id=11

[3] M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza, "TUIO – A Protocol for Table-based Tangible User Interfaces", In Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW'2005), Vannes (France), 2005, http://www.tuio.org

[4] "Comment les caméras couleur fonctionnent", livre blanc du constructeur The Imaging Source, 2005, http://www.theimagingsource.com/downloads/howcolcamswp.fr_FR.pdf.

[5] J.O. Wobbrock, A.D. Wilson, Y. Li, "Gestures without libraries, toolkits or training : A $1 recognizer for user interface prototypes", In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07). Newport, Rhode Island (October 7-10, 2007). New York : ACM Press, pp. 159-168, http://depts.washington.edu/aimgroup/proj/dollar/