

Tableau virtuel interactif

Baptiste Saleil, Geoffrey Mélia, Julien Pagès, Kevin Bollini

Tuteur de projet: M. Puech

30 avril 2012

- 1 Introduction
- 2 Analyse et Conception
- 3 Bibliothèque
- 4 Application
- 5 Conclusion

But du projet :

- Proposer un sujet en lien avec nos deux formations
- Concevoir une application utilisant les mouvements de l'utilisateur (sans souris)
- Développer une bibliothèque de détection d'objet dans une image
- Exploiter cette bibliothèque pour reconnaître les mouvements de l'utilisateur
- Pouvoir écrire ou dessiner à plusieurs sur un tableau virtuel

Vidéo de présentation

Plan

- 1 Introduction
- 2 Analyse et Conception
- 3 Bibliothèque
- 4 Application
- 5 Conclusion

- 1 Introduction
- 2 Analyse et Conception**
- 3 Bibliothèque
- 4 Application
- 5 Conclusion

Choix principaux

Découper le projet en deux parties distinctes :

- une bibliothèque de suivi d'objets réutilisable
- une application avec une interface naturelle exploitant cette bibliothèque

Méthodologie :

- Se renseigner, réaliser une architecture de qualité
- Répartir le travail en fonction des compétences et formations de chacun
- Développer rapidement un prototype
- Développement incrémental en ajoutant des fonctionnalités

Organisation :

- Réunions
- Deux sous-groupes
- Partage des tâches au sein des groupes
- Décisions communes (à quatre)

Collaboration :

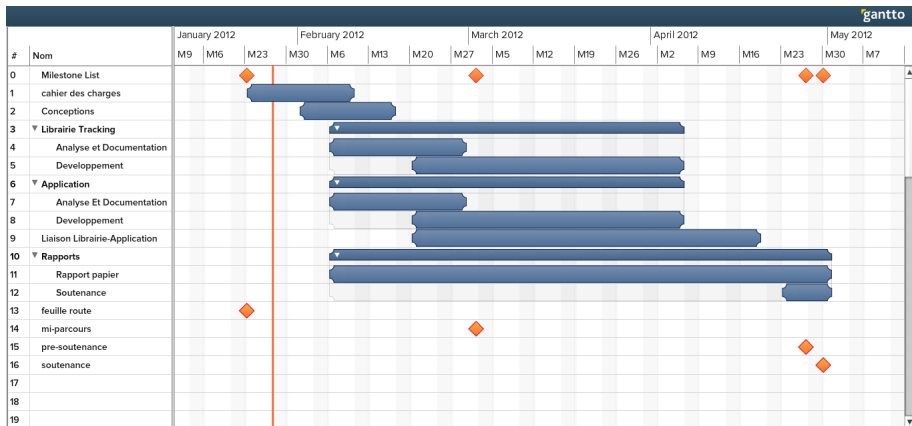
- Gestionnaire de versions (Subversion)
- Partage de documents (Mail et Subversion)
- Discussions (Mails / Instantanées)
- Édition collaborative pour le travail à distance (Gobby)

Objectifs

- Identifier les besoins et envies des utilisateurs
- Distinguer et classer les fonctionnalités de l'application
- Établir un schéma de conception dans le temps
- Faciliter le développement, avoir des buts concrets
- Produire une application réellement aboutie

Rétroplanning

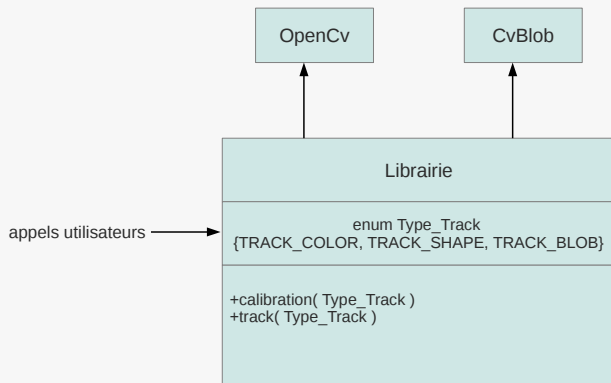
Rétroplanning (Diagramme de gantt) :



- 1 Introduction
- 2 Analyse et Conception
- 3 Bibliothèque**
- 4 Application
- 5 Conclusion

Objectifs de la bibliothèque conçue

- Distinguer complètement le suivi d'objet de l'application
- Avoir une utilisation simple sans connaissance en traitement d'image
- Permettre la détection d'actions
- Proposer un maximum de solutions de suivi
- Évaluer et comparer ces solutions



→ Utilise

Création d'une structure de données : Cursor
inclure graphique de la structure + énum

Tableau Comparatif

Méthode Caractéristique	Modèle	C. connexes	Barycentre
Vitesse	-	+	++
Précision	++	++	+
Linéarité	-	+	+
Variété curseur	++	+	++
Souplesse, adaptation	- -	+	+
Sensibilité environnement	++	-	- -
Action	Non	Oui	Oui

Table: Comparatif des différentes solutions de suivi

Deux fonctions enveloppes :

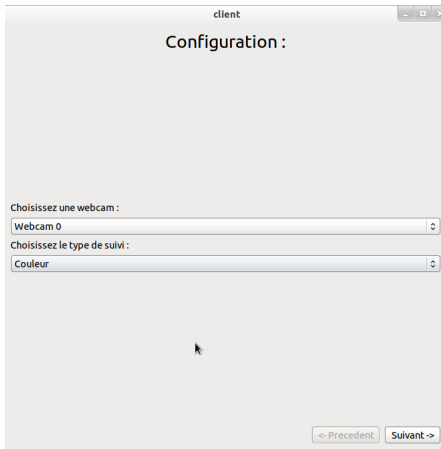
- `Cursor * calibration(IplImage * source, CvPoint A, CvPoint B, TYPE-TRACK flag)`
- `int track(IplImage * source, Cursor * oldCursor)`

Scénario type d'utilisation de la bibliothèque

La bibliothèque s'utilise en deux grandes étapes :

- Calibration, engendrant une structure Cursor
- Track, mettant à jour les informations de la structure

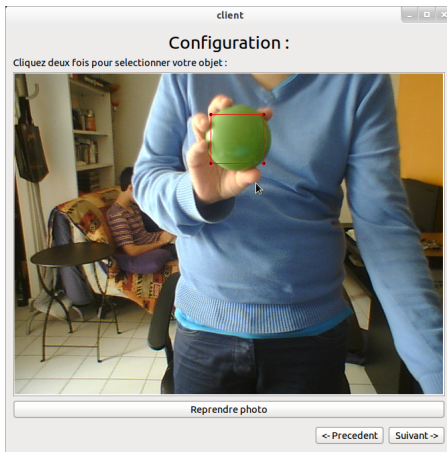
Calibration : Source d'images et TYPE_TRACK



Écran de sélection du Type_TRACK et de la source d'images

Calibration : Sélection du curseur

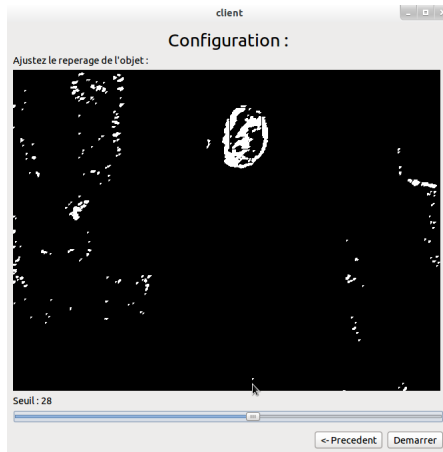
- Extrait la position de l'objet à suivre



Sélection de l'objet

Calibration couleur : Réglage du seuil

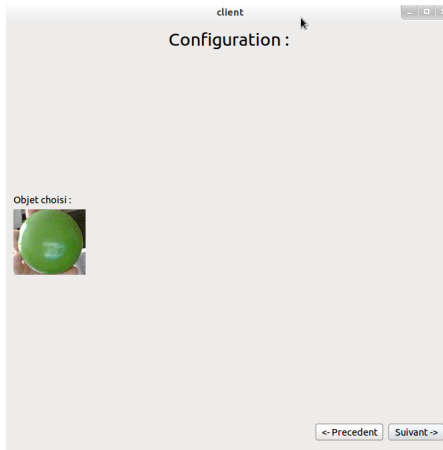
- Modifie l'attribut "threshold" de la structure Cursor.



Écran de réglage du seuil

Calibration Forme : Extraction du modèle

- Modifie l'attribut "mask" de la structure Cursor.



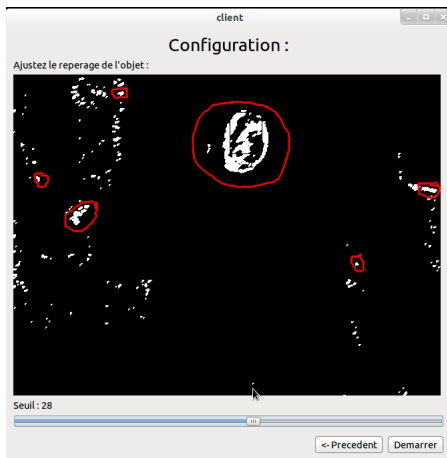
Écran de validation du modèle.

Suivi par couleur : Barycentre



Calcul du barycentre de l'image binaire

Suivi par Blob : Composantes connexes



Recherche et isolement de la composante connexe pertinente

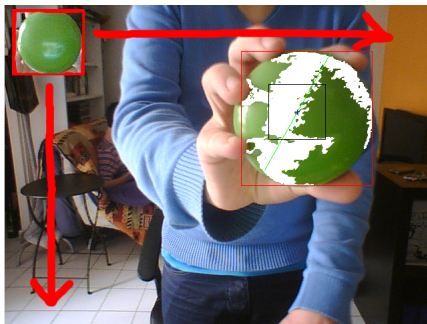
Suivi par couleur/Blob : Détection d'action

- Détection d'action par rapprochement du curseur



Retour image de l'objet suivi

- Recherche du template dans l'image



Localisation par recherche de modèle.

Objectifs atteints

- Bibliothèque utilisable et proposant plusieurs solutions de suivi
- Détection d'action implémentée dans deux des trois solutions
- Utilisation simple sans connaissances en traitement d'images

Objectifs atteints

- Bibliothèque utilisable et proposant plusieurs solutions de suivi
- Détection d'action implémentée dans deux des trois solutions
- Utilisation simple sans connaissances en traitement d'images

Difficultés

- Temps d'adaptation aux bibliothèques OpenCv et CvBlob volumineux
- Implémentation de la détection pour le suivi par modèle

Objectifs atteints

- Bibliothèque utilisable et proposant plusieurs solutions de suivi
- Détection d'action implémentée dans deux des trois solutions
- Utilisation simple sans connaissances en traitement d'images

Difficultés

- Temps d'adaptation aux bibliothèques OpenCv et CvBlob volumineux
- Implémentation de la détection pour le suivi par modèle

Ouverture

- Diversifier et optimiser les méthodes de suivi
- Rajouter des fonctionnalités côté application

- 1 Introduction
- 2 Analyse et Conception
- 3 Bibliothèque
- 4 Application**
- 5 Conclusion

- Interface intuitive
- Modulable, Extensible
- Fonctionnement transparent mode local / mode réseau
- Séparer le traitement du rendu
- Etablir un protocole simple et rapide

Architecture - Modules

Etalonnage

- Choix principaux
- Réglages

Etalonnage

- Choix principaux
- Réglages

Client

- Interface graphique
- Liens entre les modules

Etalonnage

- Choix principaux
- Réglages

Client

- Interface graphique
- Liens entre les modules

Tableau

- Dessin / Interface gestuelle
- Réseau

Etalonnage

- Choix principaux
- Réglages

Client

- Interface graphique
- Liens entre les modules

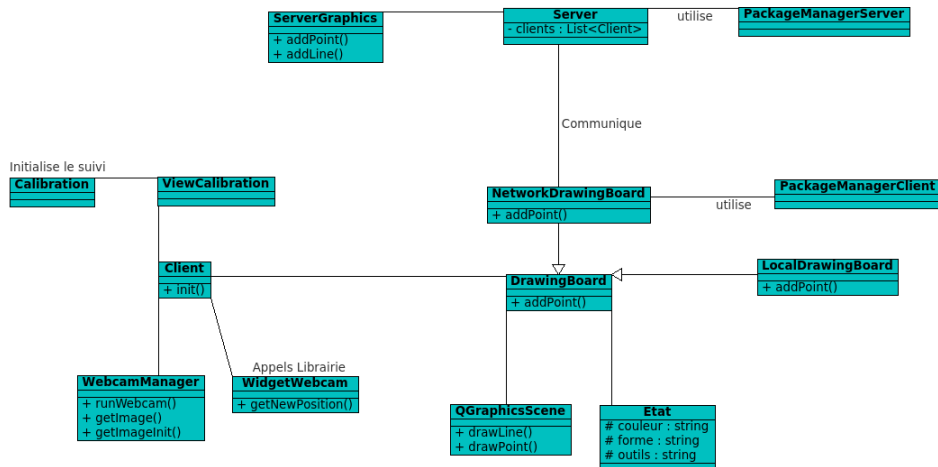
Tableau

- Dessin / Interface gestuelle
- Réseau

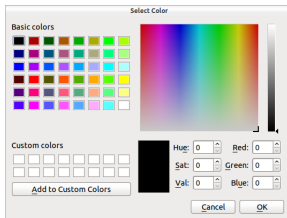
Serveur

- Communication entre clients
- Synchronisation du tableau entre les clients

Architecture - Classes

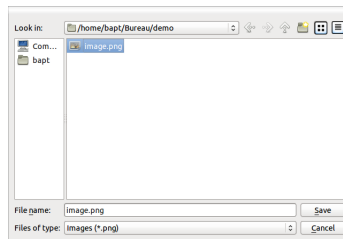
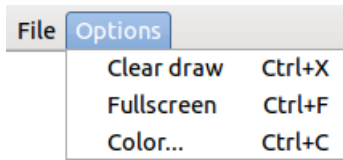


Fonctionnalités - Outils

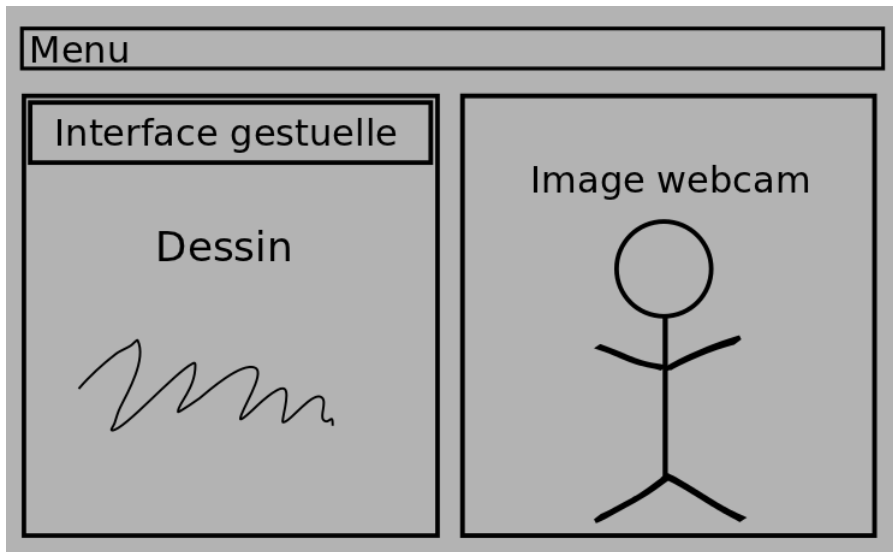


- Couleur
- Gomme
- Taille
- Affichage

Fonctionnalités - Actions



- Sauvegarde du dessin
- Vider le tableau
- Mode plein écran



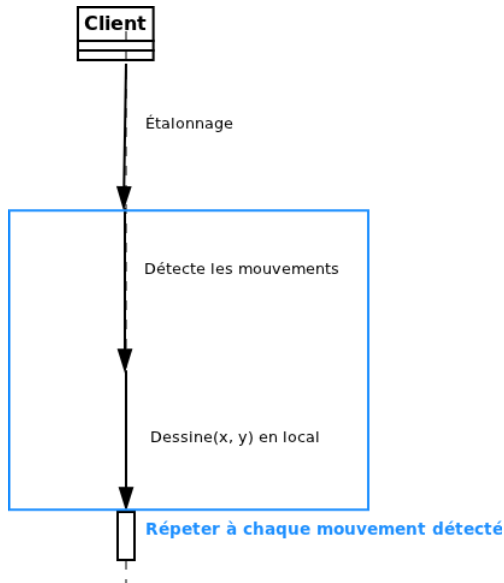
Technique

- Interface "Suivant - Précédent"
- Étalonnage obligatoire

Utilisation

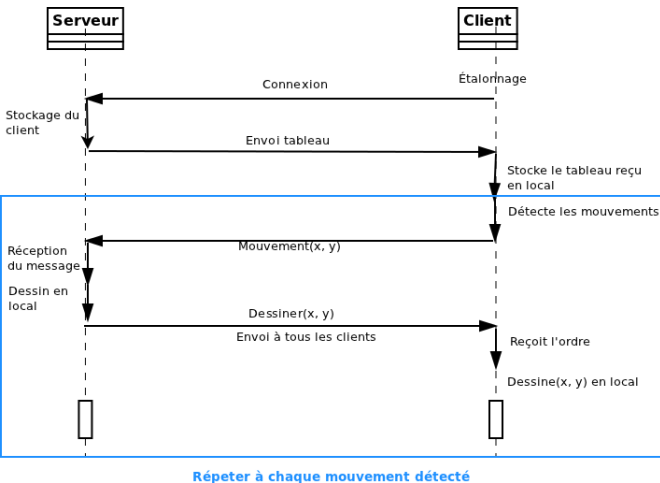
- 1 Choix webcam / Type de suivi
- 2 Sélection de l'objet
- 3 Réglage de la tolérance
- 4 Choix mode local / réseau

Fonctionnement - Local



- Étalonnage
- Détection de l'objet
- Dessin

Fonctionnement - Réseau



- Étalonnage
- Détection de l'objet
- Dessin

Pourquoi?

- Généralement oubliée
- Première expérience
- Application aboutie

Éléments

- Traduction
- Packaging (.deb)
- Documentation
- Portabilité
- Dépôt accessible
- Code propre

- Amélioration du réseau
- Amélioration des performances
- Possibilité de relancer l'étalonnage

- 1 Introduction
- 2 Analyse et Conception
- 3 Bibliothèque
- 4 Application
- 5 Conclusion**

Objectifs atteints

- Solution fonctionnelle
- Respect du cahier des charges
- Découverte (Technologies, gestion de projet...)

Conclusion

Objectifs atteints

- Solution fonctionnelle
- Respect du cahier des charges
- Découverte (Technologies, gestion de projet...)

Difficultés

- Collaboration : Développement incrémental qui oblige à beaucoup communiquer
- Formation : Traitement de l'image, Conception d'architectures
- Techniques : Architecture, gestion mémoire

Conclusion

Objectifs atteints

- Solution fonctionnelle
- Respect du cahier des charges
- Découverte (Technologies, gestion de projet...)

Difficultés

- Collaboration : Développement incrémental qui oblige à beaucoup communiquer
- Formation : Traitement de l'image, Conception d'architectures
- Techniques : Architecture, gestion mémoire

Ouverture

- Diversifier et optimiser les méthodes de suivi
- Rajouter des fonctionnalités côté application

- <http://www.sciencedirect.com.www.ezp.biu-montpellier.fr/science/article/pii/S026288561100120X>
- <http://www.irit.fr/recherches/SAMOVA/pageAnalysis.html>
- <http://www.irit.fr/~Philippe.Joly/Teaching/L3SI/ti.html>
- <http://opencv.willowgarage.com/wiki/>
- code.google.com/p/cvblob/

lien du projet :

<http://code.google.com/p/dessin-realite-augmentee/>