

# 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

## Ter de Master 1 : Tableau virtuel interactif

Tuteur de projet: W. Puech

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

- Choix de conceptions
- Gestion de projet
- Analyse
- Planning

## 3 Bibliothèque

- Architecture
- Fonctionnement
  - Calibration
  - Suivi
- Comparatifs

## 4 Application

- Module Local
- Module Réseau

## 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 :

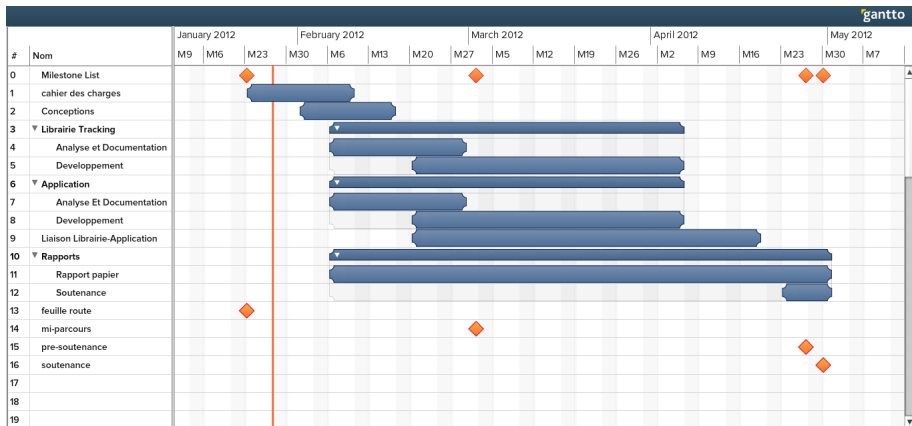
- Gestionnaire de version (Subversion)
- Partage de documents (Mail et Subversion)
- Discussions (Mails / Instantanée)
- Édition collaborative pour le travail à distance (Gobby)

## 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

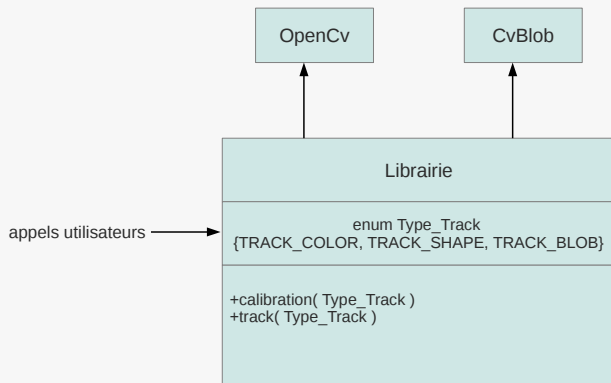
Rétroplanning (Diagramme de gantt) :



- 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 une détection d'action
- Proposer un maximum de solutions de suivi
- Évaluer et comparer ces solutions



→ Utilise

Création d'une structure de données : Cursor

```
struct Cursor {
```

- CvPoint center
- CvPoint cornerA
- CvPoint cornerB
- ...
- IplImage \*mask
- Bool active

```
}
```

Deux fonctions enveloppes :

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



# Comparatif Couleur/Forme

## Couleur

### Avantages

- Suivi rapide
- Diversité possible de curseurs

### Faiblesses

- Sensibilité à l'environnement
- Dépendant de la qualité du dispositif d'acquisition

# Comparatif Couleur/Forme

## Couleur

### Avantages

- Suivi rapide
- Diversité possible de curseurs

### Faiblesses

- Sensibilité à l'environnement
- Dépendant de la qualité du dispositif d'acquisition

## Forme

### Avantages

- Suivi moins dépendant de la qualité de l'environnement
- Efficace sur des objets 'complexes'

### Faiblesses

- Suivi lent
- Très sensible aux variations du curseur

# Comparatif Simple/composante connexe

## Barycentre simple

### Avantages

- Suivi rapide

### Faiblesses

- Sensibilité aux parasites (fausses détections)
- Précision fortement dépendante de l'environnement

## Barycentre simple

### Avantages

- Suivi rapide

### Faiblesses

- Sensibilité aux parasites (fausses détections)
- Précision fortement dépendante de l'environnement

## Barycentre composante connexe

### Avantages

- Suivi plus précis
- Résistance aux parasites

### Faiblesses

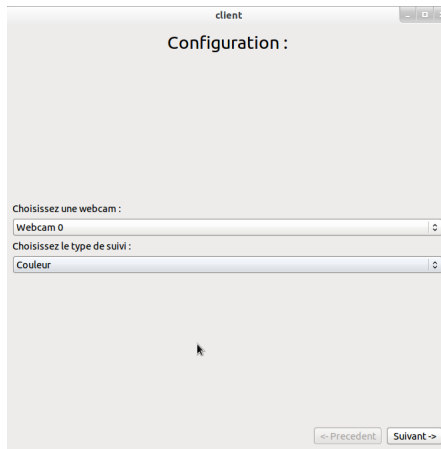
- Suivi plus lent
- Perte occasionnelle du curseur

# Scénario type d'usage 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

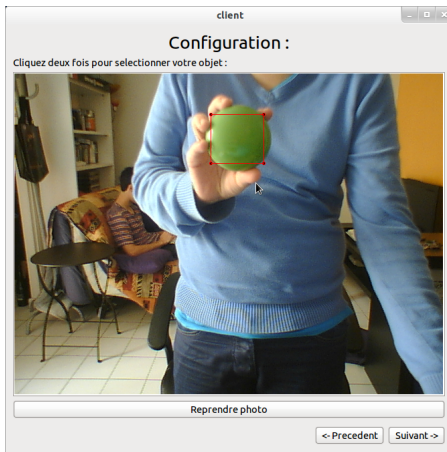


Écran de selection du Type\_TRACK et de la source d'images



# Calibration : Selection du curseur

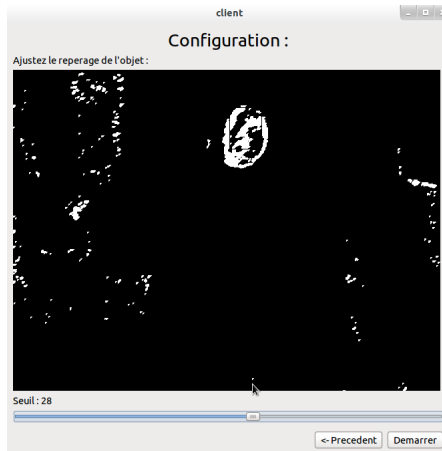
- Extrait la position de l'objet à suivre



Sélection de l'objet

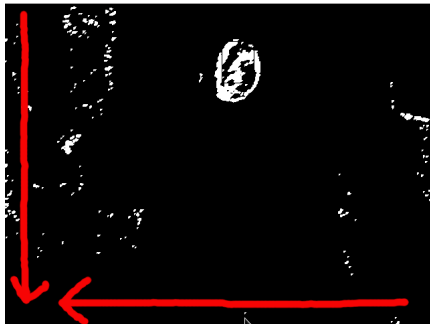
# Calibration couleur : Réglage du seuil

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



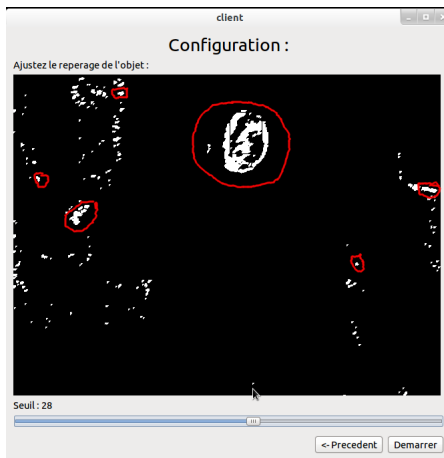
Écran de réglage du seuil

## 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

## 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 et ouverture

- Temps d'adaptation aux bibliothèques OpenCv et CvBlob volumineux
- Implémentation de la détection pour le suivi par forme
- Ajout à la bibliothèque de nouvelles fonctions de suivi

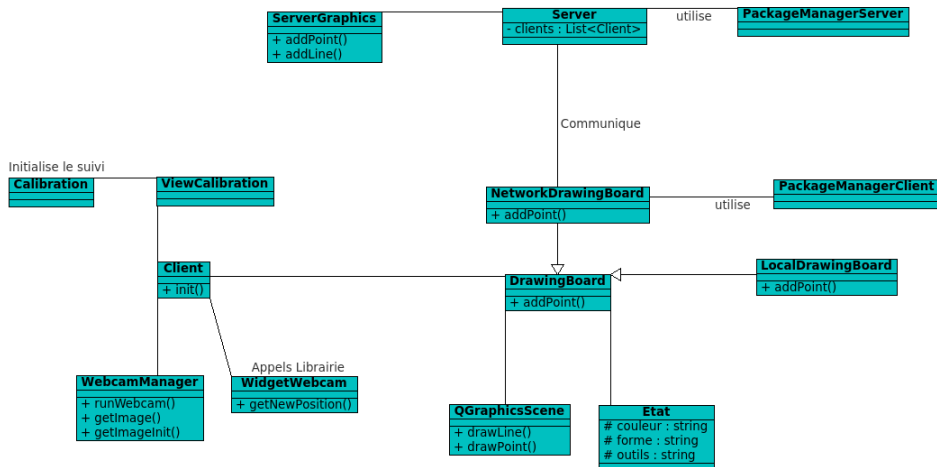


- 1 Introduction
- 2 Analyse et Conception
- 3 Bibliothèque
- 4 Application**
- 5 Conclusion

## Objectifs de l'architecture conçue

- Avoir une application modulable et facilement extensible
- Fonctionnement identique pour les classes principales en réseau ou en local
- Pouvoir rajouter facilement des outils (pinceaux, gomme, etc.)
- Séparer le traitement du rendu

# Diagramme de classes



L'application est utilisable en local et en réseau, avec un fonctionnement identique.

Le dessin virtuel interactif propose différents outils :

- La possibilité de changer la couleur du pinceau
- Celle d'ajuster la taille du pinceau
- Une gomme
- Fonctionnalité permettant d'exporter le dessin
- Possibilité de nettoyer complètement le tableau
- Mode plein-écran avec le dessin uniquement (vidéoprojecteur)
- Utilisation simultanée par plusieurs utilisateurs
- Application traduite et packagée (en .deb)

## Fonctionnement en local

- Étalonnage selon la méthode voulue, choix du mode local
- Détection d'un mouvement, dessin directement sur le tableau en respectant les options

L'étalonnage se déroule en plusieurs phases.

- Choix de la webcam et de méthode de suivi
- Choix de l'objet à suivre à partir d'une photo, en l'entourant d'un rectangle
- Réglage du seuil de tolérance à partir du retour de l'étalonnage
- Choix du mode : réseau ou local

# Utilisation de l'application

L'interface permet de visualiser le flux vidéo, et le dessin.  
Les mouvements sont détectés, et le dessin est effectué à partir de ces mouvements.

## Objectifs atteints

- Application fonctionnelle et utilisable
- Beaucoup d'outils voulus implémentés



## Objectifs atteints

- Application fonctionnelle et utilisable
- Beaucoup d'outils voulus implémentés

## Difficultés et ouverture

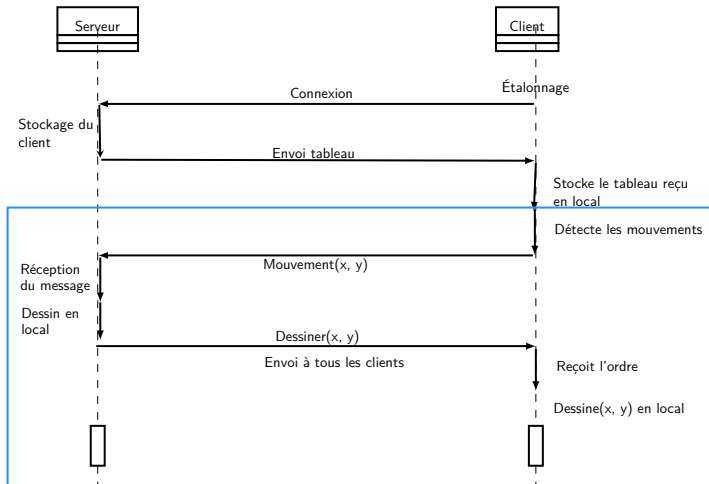
- Faire une interface gestuelle pour sélectionner gomme, couleur et forme
- Améliorer la gestion du dessin
- Relancer l'étalonnage sans relancer l'application

## Fonctionnement en réseau

- Étalonnage selon la méthode voulue, choix du mode réseau
- Récupération du dessin actuel par le client
- Détection d'un mouvement
- Envoi au serveur de ce mouvement (et des options) en respectant le protocole
- Réception du paquet côté serveur, dessin du serveur
- Envoi aux clients de ce point, avec les options (épaisseur, couleur)
- Réception côté client, et dessin en local

# Fonctionnement global : schéma

Déroulement du fonctionnement en réseau de l'application :



Répéter à chaque mouvement détecté

## Objectifs atteints

- Architecture identique au fonctionnement en local
- Application fonctionnelle en mode réseau malgré la difficulté
- Tous les outils fonctionnent

## Objectifs atteints

- Architecture identique au fonctionnement en local
- Application fonctionnelle en mode réseau malgré la difficulté
- Tous les outils fonctionnent

## Difficultés et ouverture

- Optimiser fortement le mode réseau pour réduire les problèmes de lenteur
- Mettre par défaut une couleur à chaque utilisateur

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

# 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, Fuites de 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, Fuites de 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/>