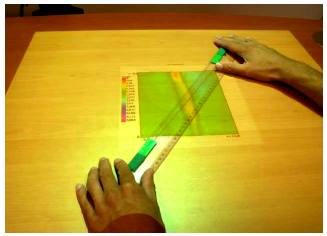
GeoTUI



www.estia.fr/~geotui

Implémentation du prototype de GeoTUI

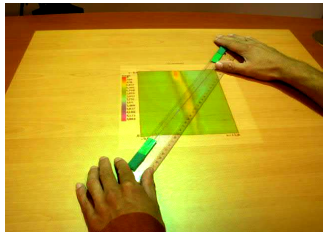
GT Réalité Mixte

LIPSI, ESTIA, Bidart

Jeudi 3 Mai 2007

ESTIA
RECHERCHE-LIPSI
CCI BAYONNE PAYS BASQUE

GeoTUI

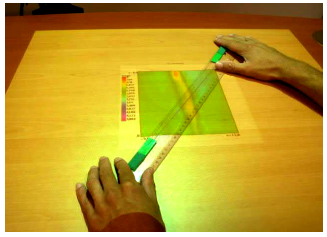


www.estia.fr/~geotui

Introduction

- Guillaume Rivière
- "Interaction et visualisation en Géosciences"
  - Thèse de doctorat (2008)
  - Bourse MESR

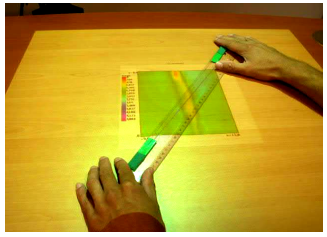
Nadine Couture	Maylis Delest
I3D	InfoViz
LIPSI	LaBRI



Collaboration de l'IFP

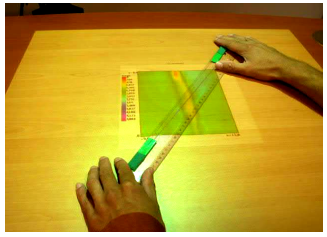
- 05/05/2004 Réunion CREATI
  - Nadine Couture démarche Jacques Jacobs

- Février - Juin 2005 Mémoire de Master
  - "Faisabilité d'une Interface Tangible pour la Validation d'Hypothèses en Géosciences"



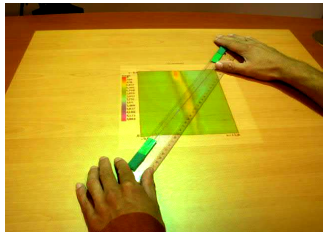
Problématique

- Travail collaboratif co-présent
  - Salle Immersive de RV
    - HMD ou Data Gloves
    - Cher et volumineux (salle dédiée)
    - Unique
    - Planning
- Tâche métier complexe
  - Ecran, souris, clavier
    - Inadaptés ou insuffisants



Objectif

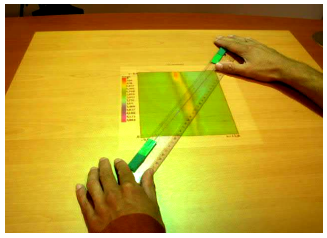
- Une Interface Tangible (TUI)
  - Simple
  - Déployable
  - Retour aux méthodes traditionnelles
  - Faciliter les tâches



Quelle tâche ?

- Sélectionner une coupe de sous-sol
- Tâche articulatoire
  - Naviguer dans le cube
  - Construire un modèle
  - Tracer des splines
  - Attribuer des propriétés physiques

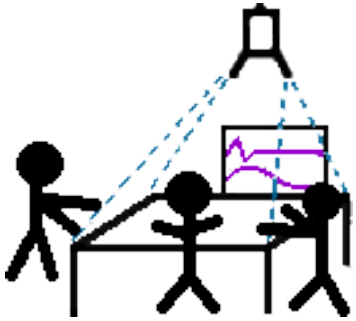
GeoTUI

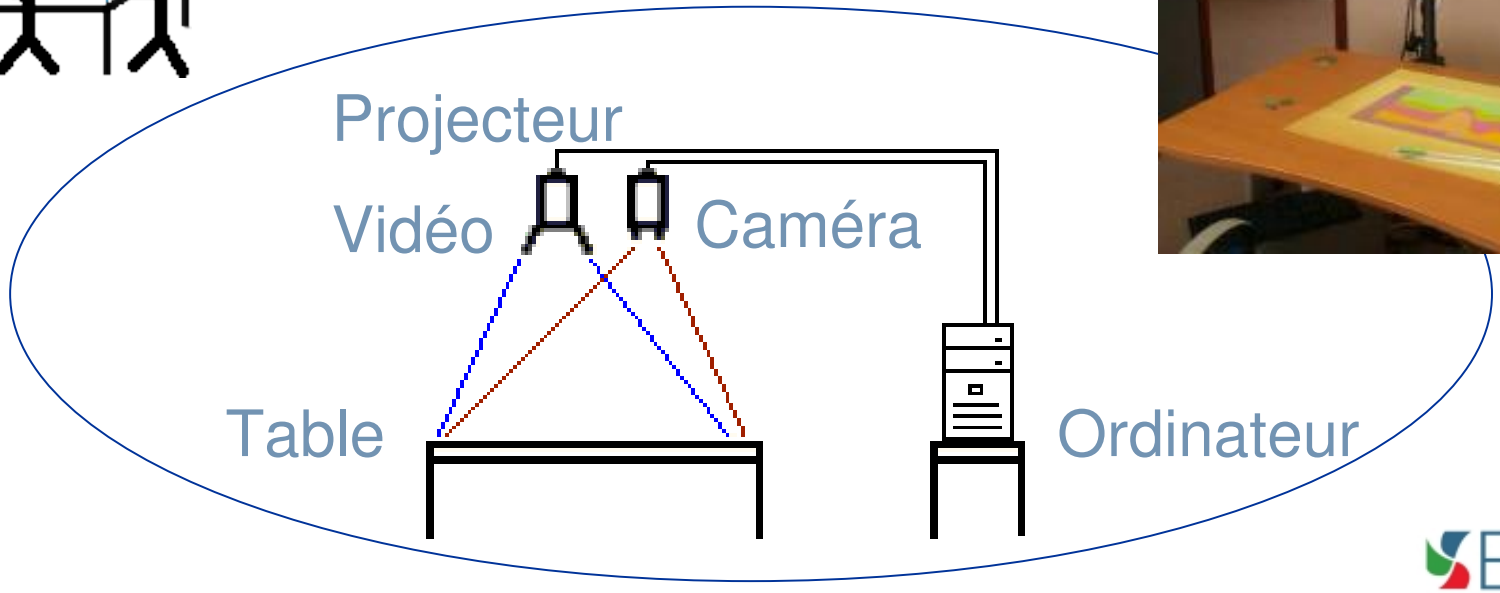


www.estia.fr/~geotui

Conception de GeoTUI

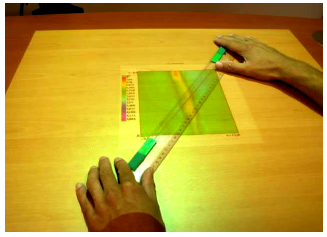
- Interagir autour d'un TableTop





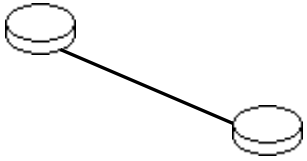
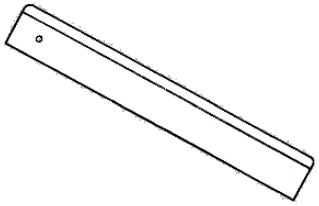
ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE

GeoTUI

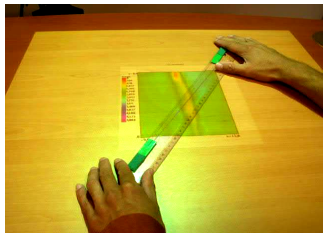


www.estia.fr/~geotui

Conception de GeoTUI

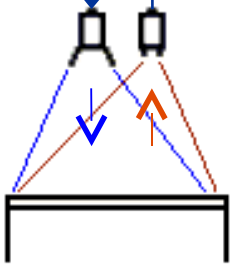
- Quel interacteur pour réaliser une coupe ?
  - 1 palet 
  - 2 palets 
  - 1 règle 

ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE



www.estia.fr/~geotui

Prototype de GeoTUI



capture

Vision

Extraction de vert

get

GeoTUI
C++

JOHN
JAVA

requête

Contrôleur de dialogue

Noyau fonctionnel

image

demande

set

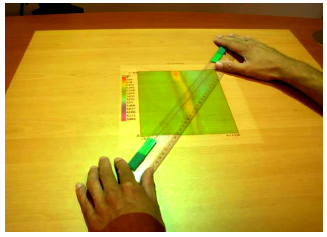
key

projection

Présentation

GTKmm

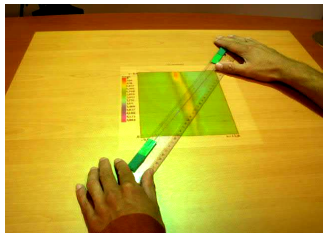
Client | *Connexion socket* | *Serveur*



GeoTUI | JOHN

- Communication via socket Unix
- Protocole : Requête / Image
- JOHN se comporte comme un serveur
- GeoTUI se comporte comme un client

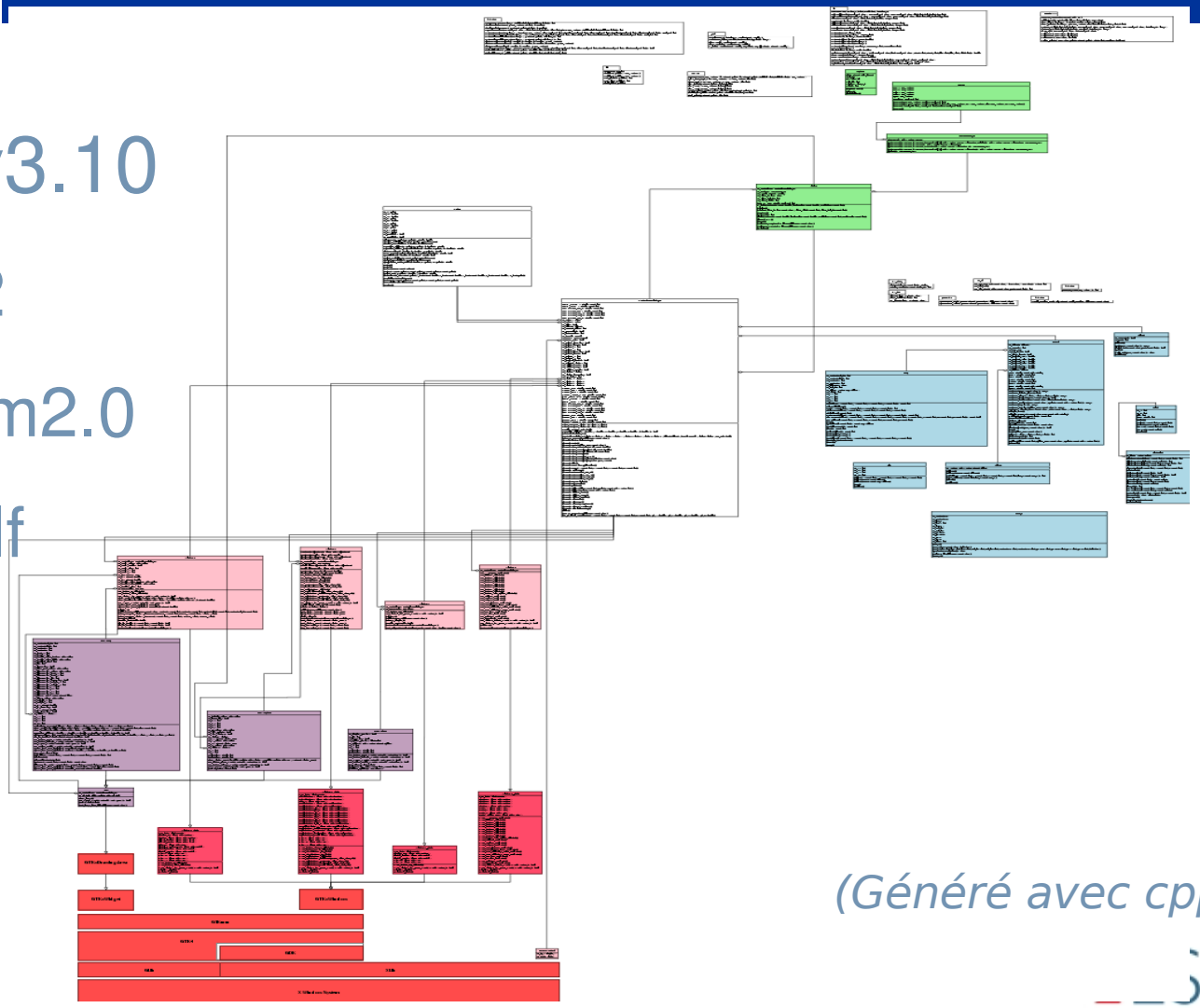
GeoTUI



www.estia.fr/~geotui

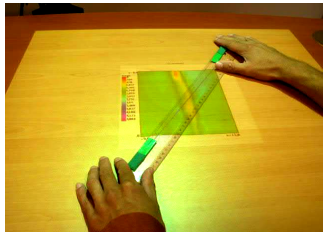
Diagramme de classes

- GeoTUI v3.10
  - Glade2
  - GTKmm2.0
  - Gandalf
  - V4L



(Généré avec cpp2dia)

ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE



Références

- **V4L2** video capture API for Linux
  - http://linux.bytesex.org/v4l2/
- **Gandalf** vision library
  - http://gandalf-library.sourceforge.net/
- **cpp2dia** reverse engineering
  - http://cpp2dia.sourceforge.net/

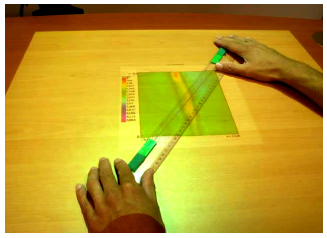
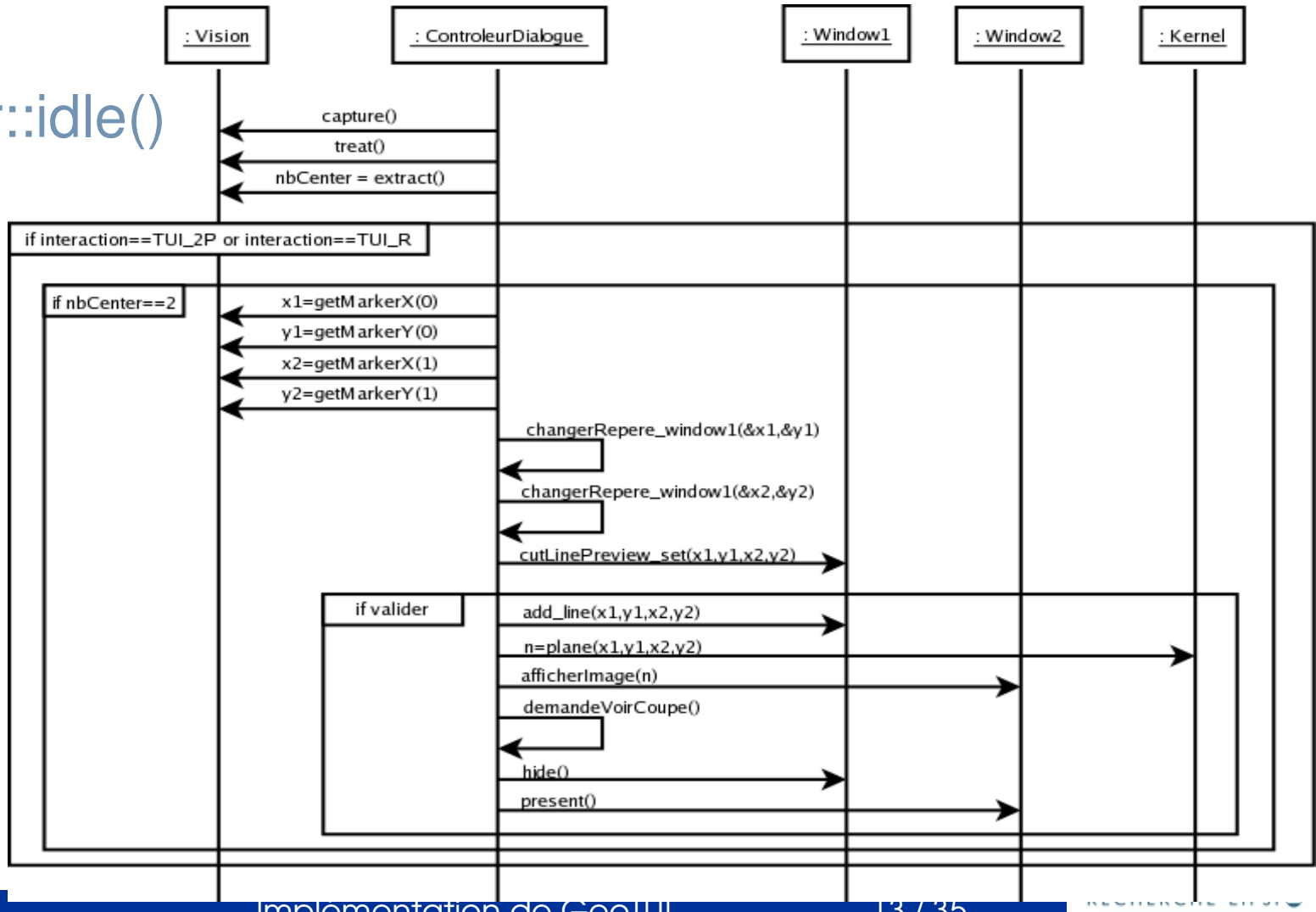
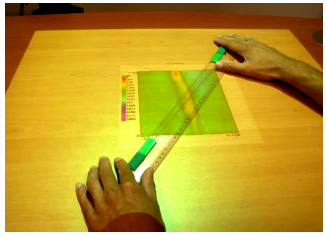


Diagramme de séquence

- Controleur::idle()

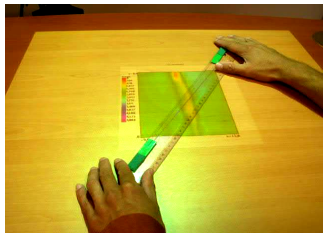




Redressement

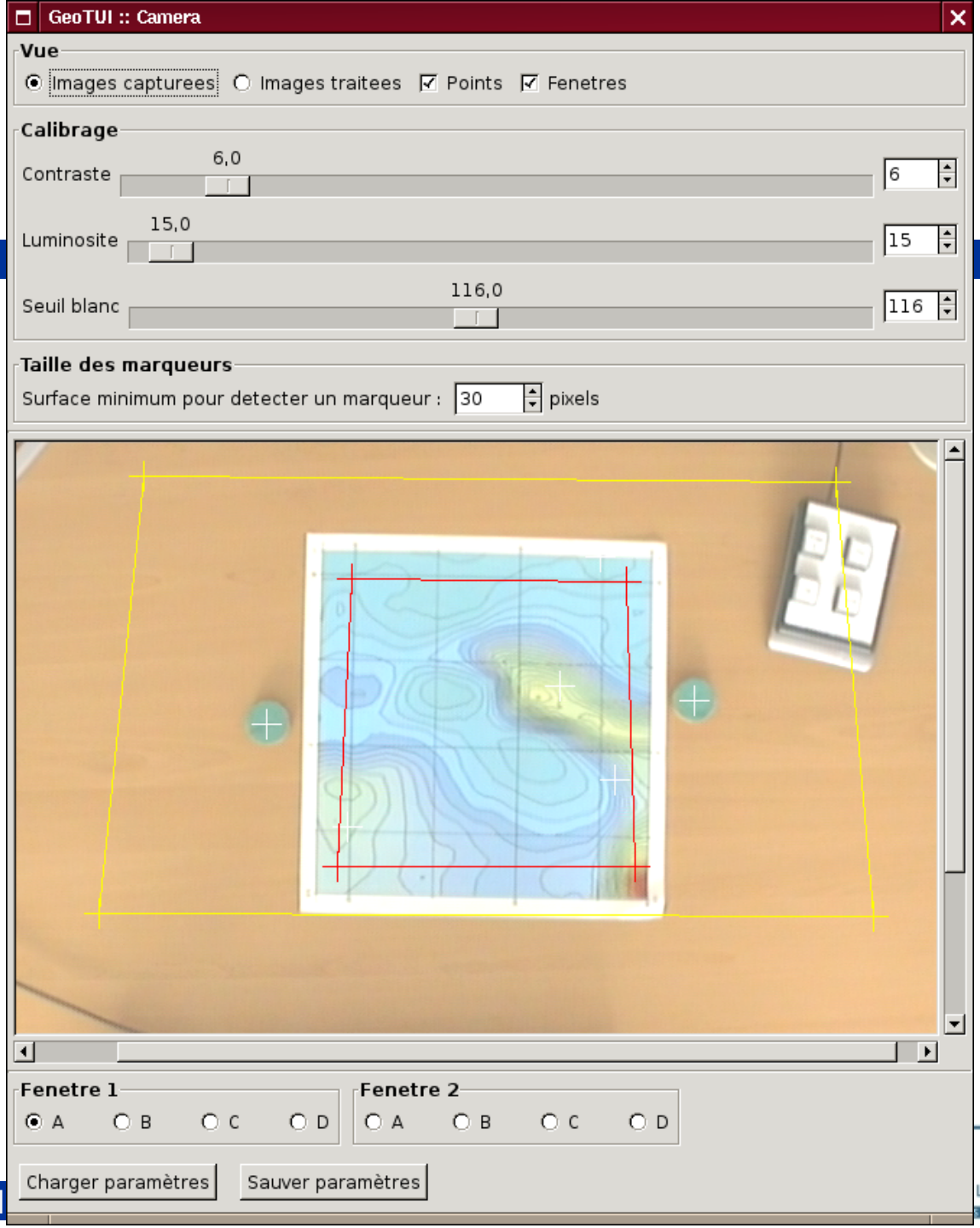
- Coordonnées dans l'image capturée
  - Zone rectangulaire déformée et inclinée
- Coordonnées dans la carte
  - Zone rectangulaire

GeoTUI

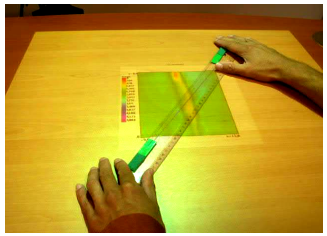


www.estia.fr/~geotui

- Fenêtre de réglage

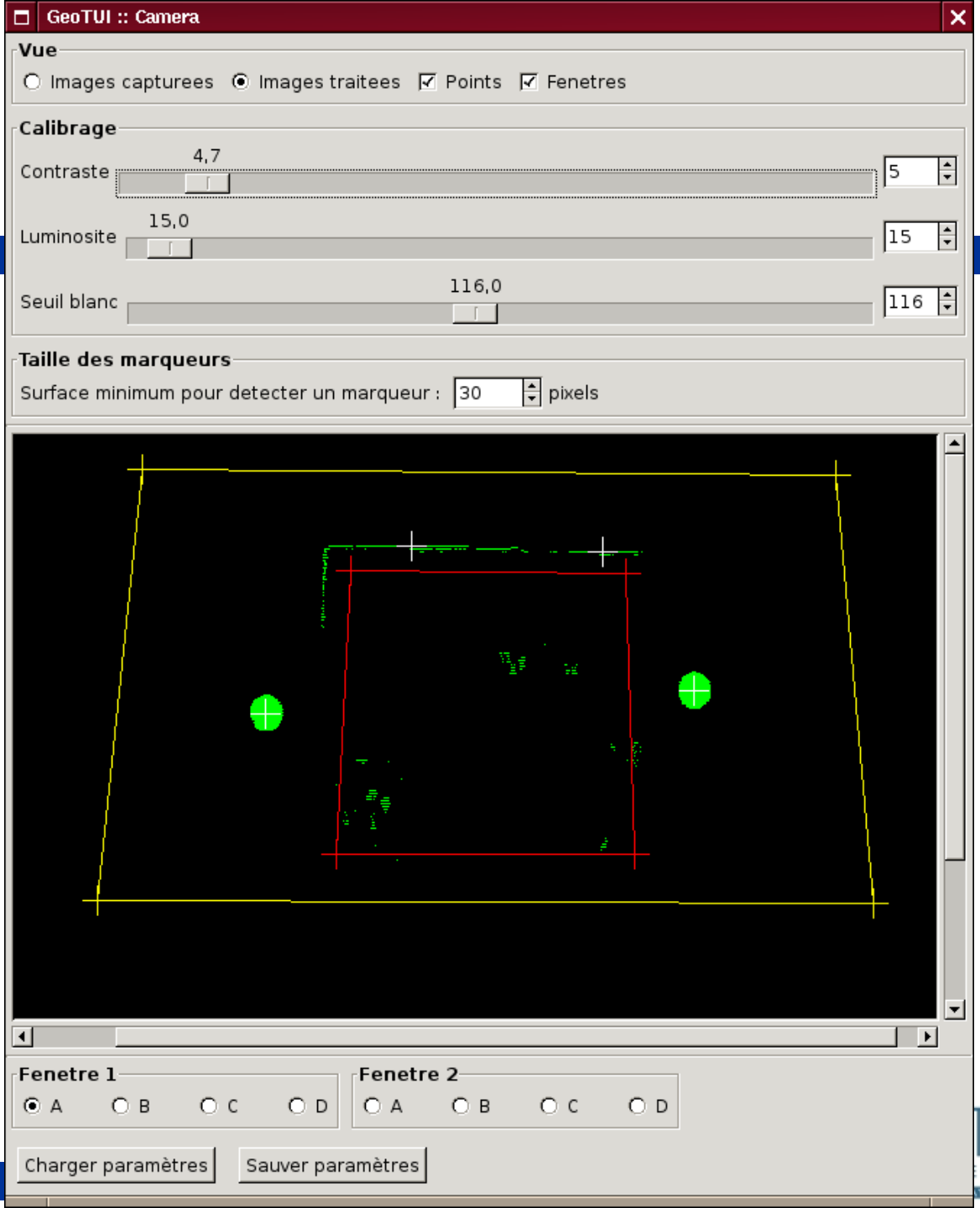


GeoTUI

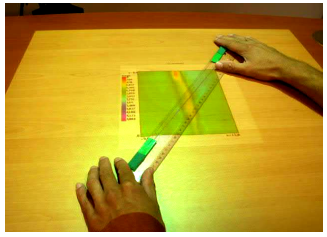


www.estia.fr/~geotui

- Fenêtre de réglage



GeoTUI



www.estia.fr/~geotui

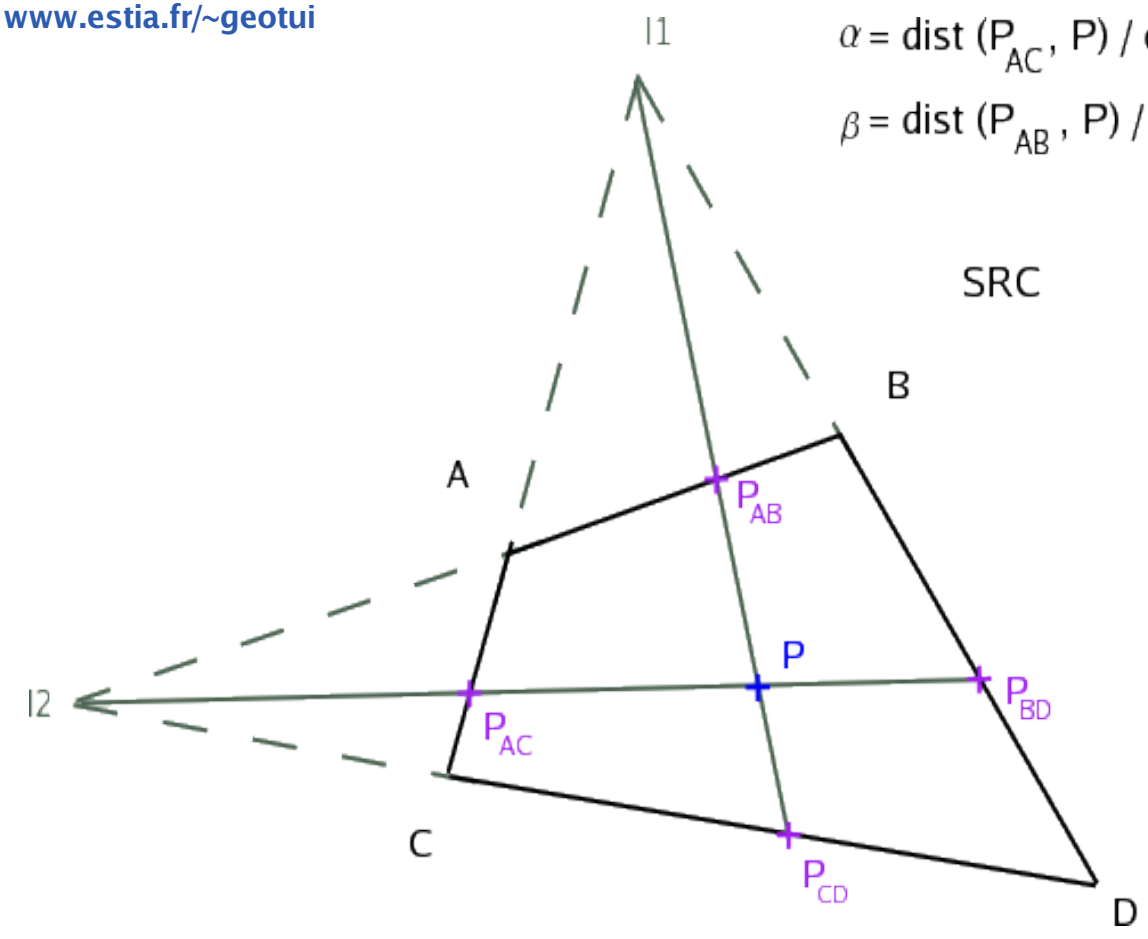
Redressement

$\alpha = \text{dist}(P_{AC}, P) / \text{dist}(P_{AC}, P_{BD})$

$\beta = \text{dist}(P_{AB}, P) / \text{dist}(P_{AB}, P_{CD})$

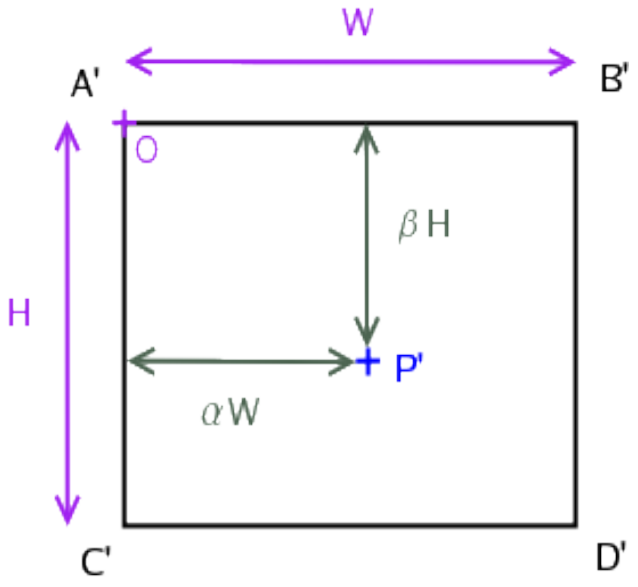
$P'_x = O_x + \alpha W$

$P'_y = O_y + \beta H$

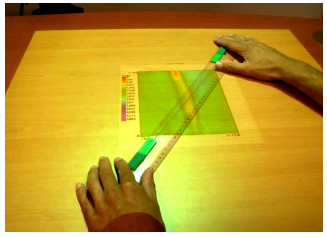


SRC

DST



ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE

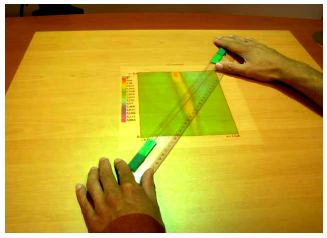


Redressement

- Types utilisés :

```
struct point {
        double x ;
        double y ;
} ;
```

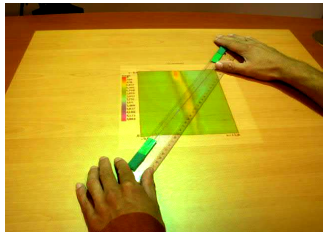
```
/* Si vertical est faux alors l'equation
 * affine y = a * x + b est définie,
 * sinon la valeur en abscisse x est définie.
 */
struct droite {
  double a ;
        double b ;
        bool   vertical ;
        double x ;
} ;
```



Redressement

- Fonctions utilisées :

```
droite equation_affine (point p1, point p2);

point  projection_orthogonale (droite d, point p);

double distance (point p1, point p2); // Distance euclidienne

point  intersection (droite d1, droite d2);

bool   paralleles (droite d1, droite d2);

/* Produit scalaire des vecteurs orientés U=(Ud,Uf) et V=(Vd,Vf) */
bool   produit_scalaire (point Ud, point Uf,
                         point Vd, point Vf);
```



Redressement

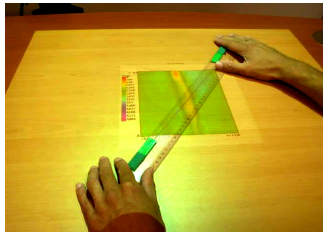
- Pseudo-code du programme

```
/* Coordonnées du cadre délimitant
 * la carte dans l'image capturée.
 */
point A ← ...;
point B ← ...;
point C ← ...;
point D ← ...;
```

```
/* Variables calculées en fonctions
 * des coordonnées du cadre.
 */
droite AB ← equation_affine (A, B);
droite CD ← equation_affine (C, D);
droite AC ← equation_affine (A, C);
droite BD ← equation_affine (B, D);

point  I1 ← intersection (AC, BD);
point  I2 ← intersection (AB, CD);
```

```
point redresser (point P, // Point P exprimé dans le cadre (A,B,C,D).
                 point O, double W, double H)
debut

  point res ; // Point P' exprimé dans le cadre (O, W, H).
  point P_AB, P_CD, P_AC, P_BD ;

  /* Calcul du rapport beta */
  si paralleles (AC, BD) alors
    P_AB ← projection_orthogonale (AB, P);
    P_CD ← projection_orthogonale (CD, P);
  sinon
    droite PI1 ← equation_affine (P, I1);
    P_AB ← intersection (PI1, AB);
    P_CD ← intersection (PI1, CD);
  finsi

  double percentY ← distance (P_AB,P) * 100 / distance (P_AB,P_CD);

  si non (produit_sclaire (P_AB, P, P_AB, P_CD) > 0) alors
    percentY ← percentY * -1 ; // vecteurs en sens opposé
  finsi
```

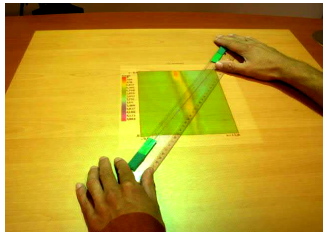
```
  /* Calcul du rapport alpha */
  si paralleles (AB, CD) alors
    P_AC ← projection_orthogonale (AC, P);
    P_BD ← projection_orthogonale (BD, P);
  sinon
    droite PI2 ← equation_affine (P, I2);
    P_AC ← intersection (PI2, AC);
    P_BD ← intersection (PI2, BD);
  finsi

  double percentX ← distance (P_AC,P) * 100 / distance (P_AC,P_BD);

  si non (produit_scalaire (P_AC, P, P_AC, P_BD) > 0) alors
    percentX ← percentX * -1 ; // vecteurs en sens opposé
  finsi

  /* Calcul du résultat et fin de la fonction */
  res.x ← O.x + percentX * W / 100 ;
  res.y ← O.y + percentY * H / 100 ;

  retourner res ;

finredresser
```



Redressement

- Démonstration (petit logiciel de test)
- Tout le code est encapsulé dans une classe Cadre.cc réutilisable.

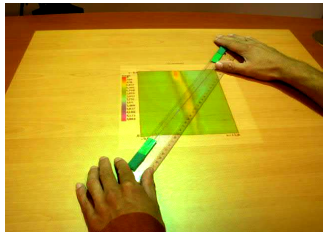
```
class Cadre {
  class point { ... } ;
  class droite { ... } ;
  void setZone (point A, point B, point C, point D) ;
  void redresser (point Psrc,
                  point O, double W, double H, point *Pdst) ;
};
```



Traitement

- Extraction de vert

```
fonction extractGreen2 (in out image, in error2) retourne rien
debut
   error  ← error2 - 100
   pour chaque pixel (R, G, B) de image faire
      si G > 70 et G - error > R et G – error/2 > B alors
         pixel ← (0, 255, 0) // Le pixel est vert
      sinon
         pixel ← (0, 0, 0) // Le pixel est noir
      finsi
   finpour
fin
```



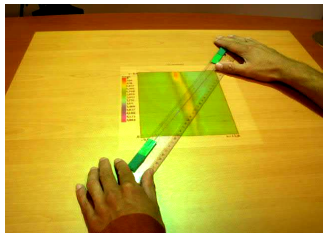
Mesure du logiciel

- Deux principaux besoins : évaluer et prédire
  - Evaluer les coûts et les moyens afférents
  - Evaluation de la qualité technique du logiciel

Entité	***Attribut***	***Métrique***
Code Source	Lisibilité Réutilisabilité Maintenabilité Etc.	Nombre de ligne de code (LOC)
	Lisibilité Réutilisabilité Maintenabilité Etc.	Pourcentage de commentaires
Tests	Complexité (vue comme la difficulté de tester un programme)	Nombre cyclomatique de McCabe v(G) (1976)
	Nombre d'erreurs	LOC

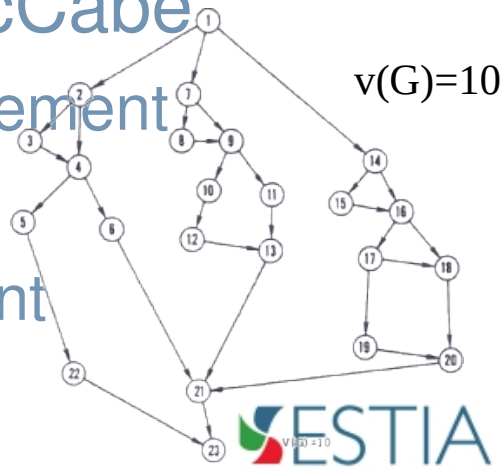
[Habrias94]

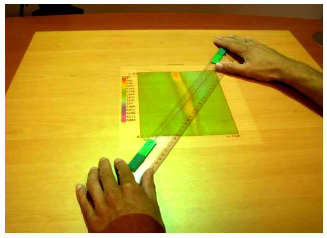
(Extrait du cours de F.Julliard "Qualimétrie des logiciels", ENIB, 05-06)



Mesure du logiciel

- Métriques
  - LOC : Lines Of Code
    - LOCphy *(total)*
    - LOCpro *(code)*
    - LOCcom *(commentaires)*
    - LOCbl *(lignes vides)*
  - v(G) : Nombre Cyclomatique de McCabe
    - v(G) est le nombre de chemins linéairement indépendant dans l'organigramme.
    - v(G) = 1 pour un programme consistant en seulement des états séquentiels.

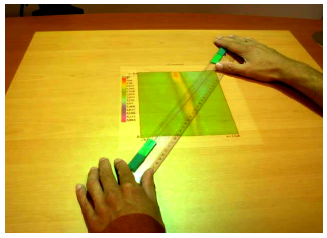




Mesure du logiciel

- CMAXDENS : Densité de complexité [Gil91]
  - *Average McCabe complexity per thousand lines of code*
    - "Normalisation" de la compléxité cyclomatique
    - Bonne corrélation avec l'effort de maintenance
  - CMAXDENS = v(G) / KNCSLOC
  - NCSLOC : Non Comment Source Line Of Code
    - "Toutes les lignes, sauf commentaires et lignes blanches. Inclut en-têtes de programme ou de modules, toutes les déclaration exécutables ou non, et les instructions à l'usage du compilateur."

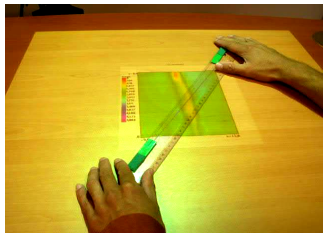
(Extrait du rapport de P. Dugerdil "Mesure de complexité et maintenance de code", HEG, Mai 05)



Mesure du logiciel

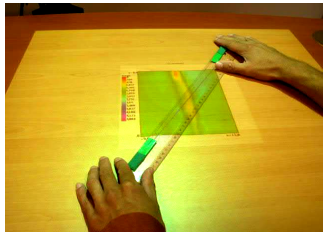
- Recommandations : (http://www.verifysoft.com)
  - LOC
    - Commentaires : 30% à 75%
    - Fonctions : 4 à 40 lignes
    - Fichiers : 4 à 400 lignes
  - Complexité Cyclomatique
    - Fichier : v(G) < 100
    - Fonction / Méthode : v(G) < 15

"Si une fonction a un nombre cyclomatic de 15, il y a au moins 15 (mais probablement plus) chemins d'exécution dans son contenu. Plus de 15 chemins sont difficiles à identifier et tester."



Mesure du logiciel

- Utilitaires sous GNU/Linux :
  - **lc2** (C/C++ Source line counter)
    - Marick (lc, C, 1983) & Rizzuto (lc2, C/C++, 2004)
    - Affiche résultats dans terminal
  - **cccc** (C and C++ Code Counter)
    - Tim Littlefair (Partie de sa thèse de PhD, 2001)
    - Génère pages web complètes
      - Bonne approximation du nombre cyclomatique
      - Autres mesures (Object Oriented Design)



Mesure du logiciel

- GeoTUI v3.10

lc2

	Total lines	Code	Comment	Code & Comment	Blank lines
.cc	11 521	7 538	1 682	107	2 194
.hh	4 021	2 164	663	53	1 141
.cc & .hh	15 542	9 702	2 345	160	3 335

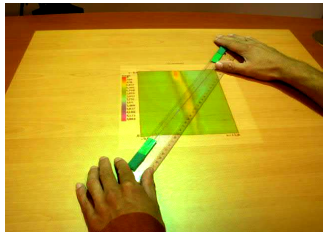
	Code	Comment	Code & Comment	Blank lines
.cc	65%	15%	1%	19%
.hh	54%	16%	1%	28%
.cc & .hh	62%	15%	1%	21%

cccc

- Nombre de modules : 60
- v(G) = 965
- v(G) par module = 16

- KNCSLOC = 9,862
- CMAXDENS = 98

GeoTUI



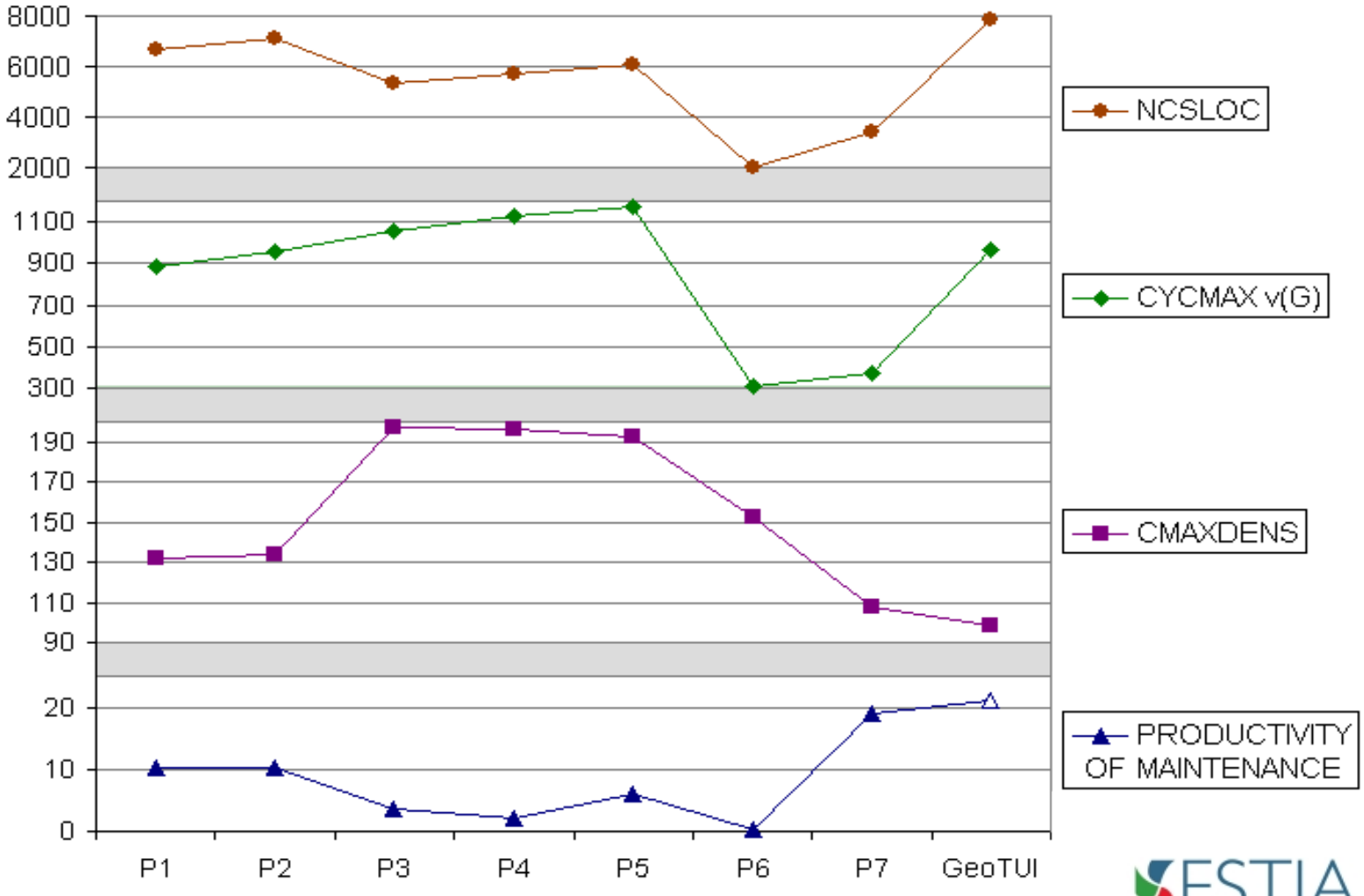
www.estia.fr/~geotui

Mesure du logiciel

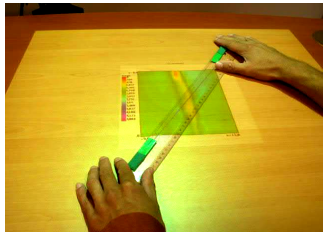
Graphique réalisé à partir des résultats obtenus sur 7 projets étudiés dans [Gil91].

Corrélation entre *productivité de maintenance* et *densité de complexité cyclomatique*.

Extrapolation pour GeoTUI de la productivité de maintenance.



ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE

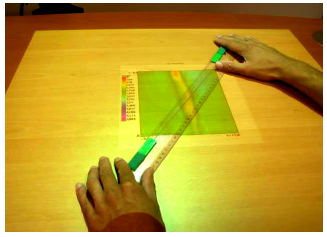


Evaluation

- But : Evaluer l'interacteur règle pour couper
  - 10 testeurs (ifp), 2 observateurs, 1 superviseur
  - Comparaison GUI vs TUI
  - Scénarios / exercices

24/07/2006

GeoTUI



www.estia.fr/~geotui

Evaluation

- Répartition des testeurs

	GUI	TUI
$U_1 = ½ U$	sc1, sc2	sc1, sc2

	TUI	GUI
$U_2 = U \setminus U_1$	sc1, sc2	sc1, sc2

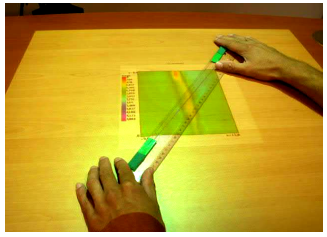
- Protocole expérimental

Formulaire personnel	Installation au poste de travail	Scénario 1 «F aites 1					Scénario 2 «C herchez les impacts dans le modèle»	Debriefing "à chaud"
		y=3 km	x=7 km	NW	½N ¼O	2 orth	10 min maximum	

- Filmé
- Traitement statistique en cours

ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE

GeoTUI

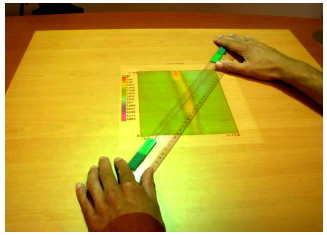


www.estia.fr/~geotui

Evaluation

- Premiers retours :
  - Choix de l'interacteur règle : 100% des testeurs
  - « C'est simple. »
  - « C'est du concret, on se trompe pas. »
  - « C'est disponible quand ? »

GeoTUI

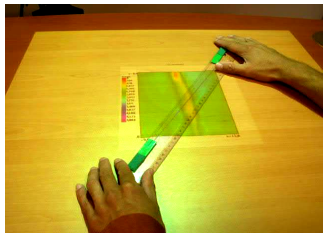


www.estia.fr/~geotui

Démonstration

- Vos questions…
- Film (2min30) ?
- http://www.estia.fr/~geotui/

GeoTUI



www.estia.fr/~geotui

Film (2min30)

UbiMob'06, Paris,5-8 Septembre 2006

GeoTUI

http://www.estia.fr/~geotui/

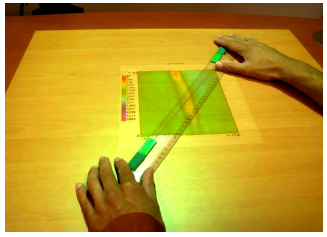
Une Interface Tangible pour la Validation d'Hypothèses en Géosciences

Nadine Couture
Guillaume Rivière

ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE

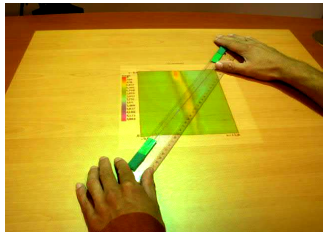
en collaboration avec IFP

ESTIA RECHERCHE-LIPSI CCI BAYONNE PAYS BASQUE



Références

- Mesure du logiciel
  - T. McCabe, 1976, "A Complexity Measure" IEEE Transactions on Software Engineering, p308-320.
    - http://www.literateprogramming.com/mccabe.pdf
  - G.K. Gil, C.F. Kemerer, 1991, "Cyclomatic Complexity Density and Software Maintenance Productivity" IEEE Trans. on Software Engineering



Références

- Mesure du logiciel
  - H. Habrias, 1994, "La mesure du logiciel", Editions Teknea.
  - P. Dugerdil, Mai 2005, Rapport HEG, "Mesure de complexité et maintenance de code"
    - http://lgl.isnetne.ch/isnet72/Phase3/complexiteCode.pdf
  - F. Julliard, 2005-2006, Cours ENIB, "Qualimétrie des logiciels"
    - http://www.enib.fr/~julliard/docs/quali.pdf