

# ENSIIE2 — Option IRA

## Réalité Augmentée — Tour d'horizon et techniques

Jean-Yves Didier

`didier@ufrst.univ-evry.fr`

- 1 Généralités
  - Définitions
  - Chronologie, paysage et applications
- 2 Systèmes de RA
  - Partie matérielle
  - Architecture globale des traitements
  - Compétences nécessaires
- 3 Problématique et verrous
  - Localisation basée vision
  - Systèmes multi-capteurs
  - Composition des scènes de RA
    - Les difficultés
  - Les occultations réel/virtuel
- 4 Architectures logicielles

## 1 Généralités

- Définitions
- Chronologie, paysage et applications

## 2 Systèmes de RA

- Partie matérielle
- Architecture globale des traitements
- Compétences nécessaires

## 3 Problématique et verrous

- Localisation basée vision
- Systèmes multi-capteurs
- Composition des scènes de RA
  - Les difficultés
- Les occultations réel/virtuel

## 4 Architectures logicielles

# Réalité augmentée (RA) : définition

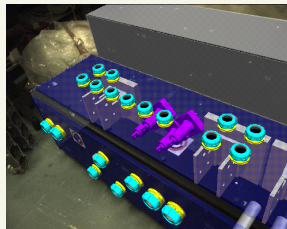
## Définition

Technique qui rend possible le **mixage d'entité virtuelles corrélées** avec des informations issues du **monde réel** dans le but d'**enrichir l'expérience utilisateur** de la réalité.

## Réalité augmentée visuelle

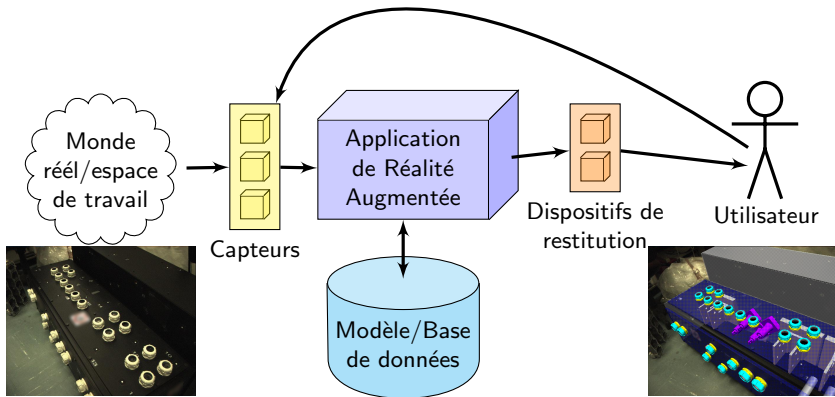
Forme de réalité augmentée la plus commune :

- Le monde réel est filmé ;
- Des images virtuelles sont incrustées.



# Système de Réalité Augmentée

## Schéma de principe général







# Fonctionnalités principales d'un système de RA

- Assurer la cohérence spatio-temporelle des augmentations ;
  - ▶ Localiser le système par rapport à son espace de travail.
- Fournir une assistance à l'opérateur ;
  - ▶ Restituer l'information augmentée à l'opérateur ;
- Traiter les commandes de l'opérateur ;
  - ▶ Proposer des interactions à l'opérateur.



## Chronologie de la RA

1966 : Ivan Sutherland, premier HMD pour RV et RA ;

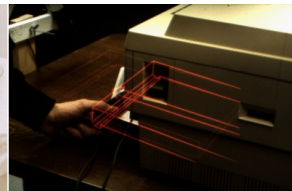
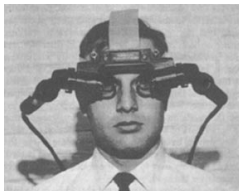
1990 : Création du terme *augmented reality*, Tom Caudell, David Mizell (Boeing) ;

1992 : KARMA, Steven Feiner ;

1999 : ARToolkit, Hirokazu Kato ;

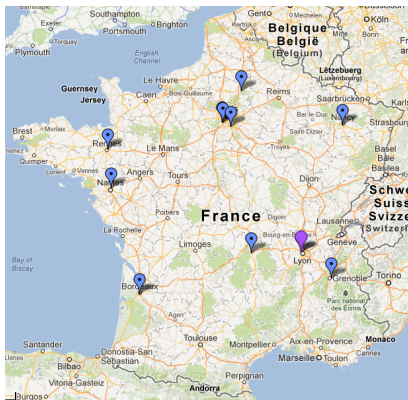
2008 : Wikitude, application de RA sur *smartphone* ;

2013 : Google glasses.





# Carte de la recherche en France en RA



## Entreprises en lien avec la RA en Europe

- Total Immersion (logiciel), Suresnes, France ;
- Laster (fabricant de lunettes), Les Ulis, France ;
- Wikitude (logiciel pour *smartphone*), Autriche ;
- Layar (solution de commerce en ligne), Pays-Bas ;
- Metaio (matériel et logiciel), Allemagne ;
- Robocortex (logiciel), Sophia Antipolis, France ;
- Diotasoft (logiciel et matériel), Massy, France.

## Applications de la RA

- Aide au geste chirurgical ;
- Valorisation du patrimoine (*cultural heritage*) ;
- Commerce virtuel ;
- Aide à la maintenance ;
- Affichage tête haute (applications militaires et civiles) ;
- Divertissement : jeux vidéo, retransmission sportives ...

# Applications de la RA

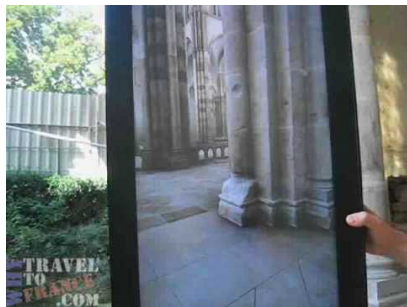
## En images



Planification, Cambridge (2005)

# Applications de la RA

## En images



Valorisation du patrimoine, Abbaye de Cluny, ENSAM (2007)

# Applications de la RA

## En images

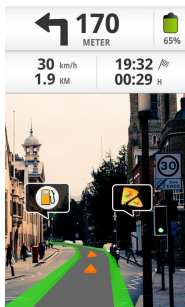


Attraction, Les animaux du futur, Futuroscope (2008)



# Applications de la RA

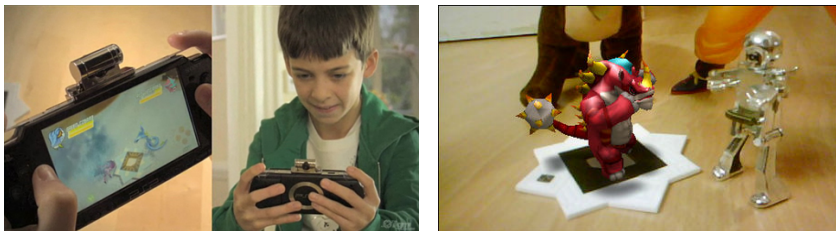
## En images



Tourisme, Wikitude (2008)

# Applications de la RA

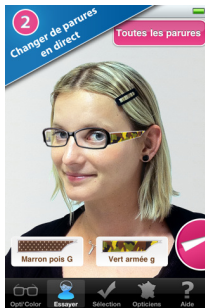
## En images



Jeu vidéo, Invizimals, Sony (2010)

# Applications de la RA

## En images



Commerce en ligne, Atol (2011)



Militaire, HUD

# Applications de la RA

## En images



Sports



Google glasses (2013)

- 1 Généralités
  - Définitions
  - Chronologie, paysage et applications
- 2 Systèmes de RA
  - Partie matérielle
  - Architecture globale des traitements
  - Compétences nécessaires
- 3 Problématique et verrous
  - Localisation basée vision
  - Systèmes multi-capteurs
  - Composition des scènes de RA
    - Les difficultés
  - Les occultations réel/virtuel
- 4 Architectures logicielles

# Capteurs et dispositifs de restitution

## Capteurs

Caméras, centrales inertielles,  
GPS, Flystick, bras haptique,  
...



## Dispositifs de restitution

Écrans, HMD, salles de réalité virtuelle, bras haptiques, ...



# Catégories de systèmes (1/2)

## Définitions

- **Réalité augmentée en vision directe (RAVD) :**
  - ▶ L'opérateur perçoit directement, le système n'effectue que le rendu virtuel, le mixage s'effectue « naturellement » ;
  - ▶ En anglais : *optical see-through augmented reality*.
- **Réalité augmentée en vision indirecte (RAVI) :**
  - ▶ L'image est filmée, traitée, puis restituée sur un écran (exocentrique).
  - ▶ En anglais : *camera see-through augmented reality*.
- **Repère égocentrique :** la vision est celle de l'utilisateur ;
- **Repère exocentrique :** la vision est celle de la caméra.

## Catégories de systèmes (2/2)

### Exemple de systèmes

- Lunettes semi-transparentes (RAVD) :
  - ▶ + main libre, – synchronisation.
- Tablette-PC, Smart phone (RAVI) :
  - ▶ + synchronisation, – main occupée.
- Réalité augmentée basée projecteur (RAVD) :
  - ▶ + main libre, synchronisation ;
  - ▶ – environnement fortement instrumenté.



## Sous-systèmes / paquetage

### Classification [McWilliams2003]

**Application** : Scénario, gestion centralisé, boucle principale ;

**Contexte** : Contexte d'utilisation ;

**Interaction** : Modalités d'entrée utilisateur ;

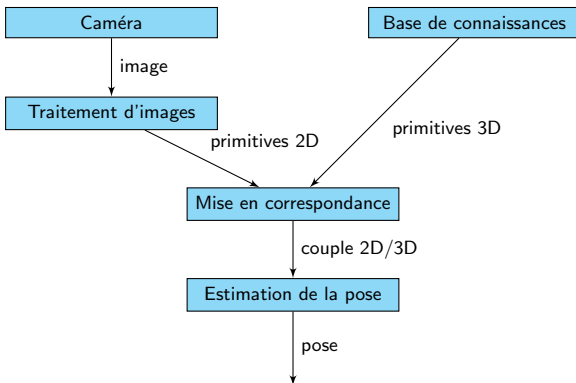
**Presentation** : Gestion des augmentations / du multi-média ;

**Suivi/localisation** : Localisation dans tous ses états ;

**Modèle du monde** : base de connaissances.

# Fonction de localisation

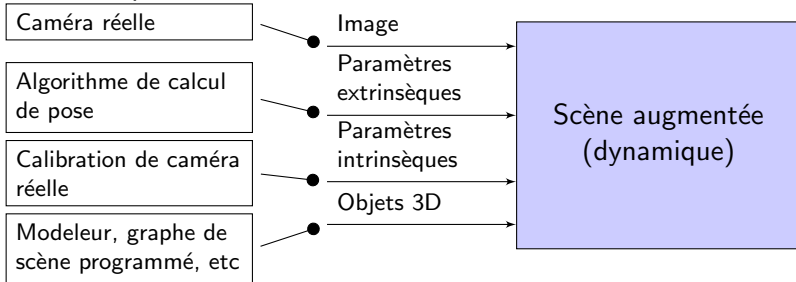
(Appliquée à la réalité augmentée basée vision)



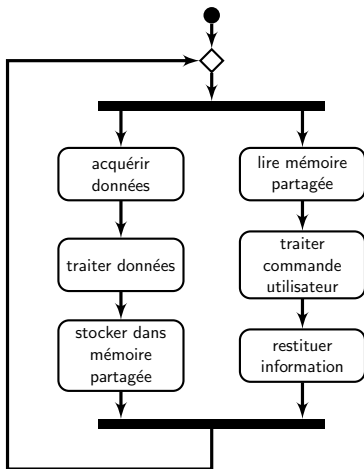
# Fonction de mixage

(Appliquée à la réalité augmentée basée vision)

Nature et provenance des données :



## Boucle principale d'un système de RA



## Disciplines/compétences nécessaires

### Une longue liste

- Traitement d'images ;

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;



# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;
- Génie logiciel / programmation ;

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;
- Génie logiciel / programmation ;
  - ▶ Capitalisation de l'expérience (reutilisabilité) ;

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;
- Génie logiciel / programmation ;
  - ▶ Capitalisation de l'expérience (reutilisabilité) ;
  - ▶ Solutions interchangeable (modularité) ;

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;
- Génie logiciel / programmation ;
  - ▶ Capitalisation de l'expérience (reutilisabilité) ;
  - ▶ Solutions interchangeable (modularité) ;
  - ▶ Standardisation des parties de logiciel.

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;
- Génie logiciel / programmation ;
  - ▶ Capitalisation de l'expérience (réutilisabilité) ;
  - ▶ Solutions interchangeable (modularité) ;
  - ▶ Standardisation des parties de logiciel.
- Réseau (applications distribuées) ;

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;
- Génie logiciel / programmation ;
  - ▶ Capitalisation de l'expérience (reutilisabilité) ;
  - ▶ Solutions interchangeable (modularité) ;
  - ▶ Standardisation des parties de logiciel.
- Réseau (applications distribuées) ;
- Concurrence & multi-threading.

# Disciplines/compétences nécessaires

## Une longue liste

- Traitement d'images ;
- Traitement du signal ;
- Interaction en réalité mixte ;
- Réalité virtuelle / synthèse d'images ;
- Génie logiciel / programmation ;
  - ▶ Capitalisation de l'expérience (réutilisabilité) ;
  - ▶ Solutions interchangeable (modularité) ;
  - ▶ Standardisation des parties de logiciel.
- Réseau (applications distribuées) ;
- Concurrence & multi-threading.

Exigences fonctionnelles

# Disciplines/compétences nécessaires

## Une longue liste

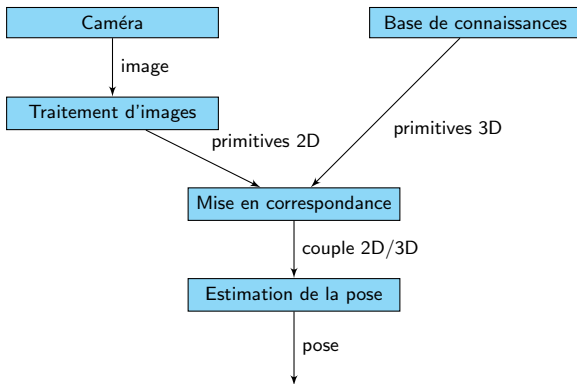
- Traitement d'images ;
  - Traitement du signal ;
  - Interaction en réalité mixte ;
  - Réalité virtuelle / synthèse d'images ;
- } Exigences fonctionnelles
- Génie logiciel / programmation ;
    - ▶ Capitalisation de l'expérience (reutilisabilité) ;
    - ▶ Solutions interchangeable (modularité) ;
    - ▶ Standardisation des parties de logiciel.
  - Réseau (applications distribuées) ;
  - Concurrence & multi-threading.
- } Exigences non-fonctionnelles



- 1 Généralités
  - Définitions
  - Chronologie, paysage et applications
- 2 Systèmes de RA
  - Partie matérielle
  - Architecture globale des traitements
  - Compétences nécessaires
- 3 **Problématique et verrous**
  - Localisation basée vision
  - Systèmes multi-capteurs
  - Composition des scènes de RA
    - Les difficultés
  - Les occultations réel/virtuel
- 4 Architectures logicielles

# Rappel : fonction de localisation

(Appliquée à la réalité augmentée basée vision)



# Définitions / Types de localisation

## Définitions

- **recalage** : technique faisant coïncider point de vue réel et point de vue virtuel en localisant le premier ;
- **suivi** : technique d'optimisation basée sur l'hypothèse de cohérence spatio-temporelle ;
- **estimation de la pose** : technique visant à obtenir la position et l'orientation de la caméra par rapport à un repère donné.

## Divers systèmes

- Utilisation de cibles codées ;
- Localisation basée modèles ;
- SLAM (*Simultaneous Localization and Mapping*) : localisation dans un environnement reconstruit en même temps.

## Rappel : estimation de la pose

### Pose = position + orientation

- Se calcule à partir de couples de points 2D/3D ;
- Minimise une erreur de projection entre points 3D transformés et points 2D de l'image ;
  - ▶ équation de la forme :

$$\operatorname{argmin}_{R,t} \sum_i \|P(RV_i + t) - v_i\|$$

- ▶ paramètres :
  - $P$  : fonction de projection ;
  - $R$  : matrice de rotation ;
  - $t$  : vecteur de translation ;
  - $V_i$  : point 3D ;
  - $v_i$  : point 2D.

# Estimation de la pose

## Un choix de diverses méthodes

- Méthodes analytiques employant de 3 à 5 points ;
- Méthodes itératives :
  - ▶ POSIT ;
  - ▶ Itération orthogonale ;
  - ▶ ....
- Méthodes spécifiques à l'objet traité (ex : cible codée).

# Le suivi

## Un problème de performances

Traiter les images dans leur intégralité prend du temps !

## Solution

- Restreindre les zones de traitement de l'image ou
- Appliquer des traitements plus simples sur l'image ;
- Exploiter la cohérence temporelle entre deux images ;
- Techniques associées : descripteurs (SIFT, SURF, ...)

# Les cibles codées

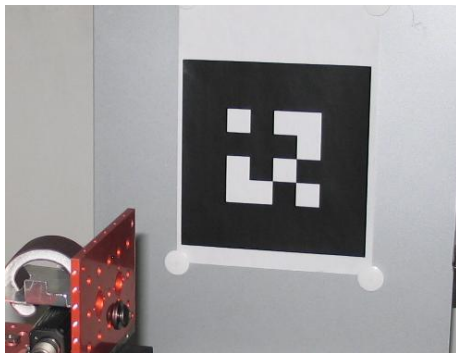
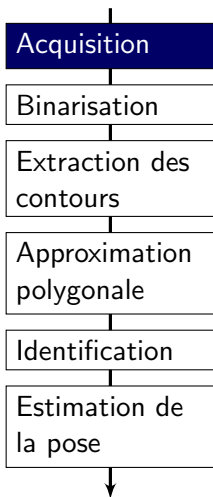
## Intérêt

- Système facile à mettre en oeuvre ;
- Peu gourmand en ressources de calcul ;
- Comporte un système d'identifiant codé ;

## Inconvénient

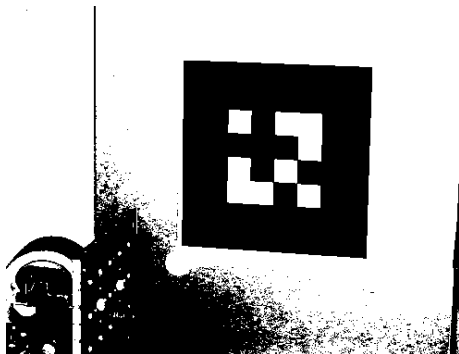
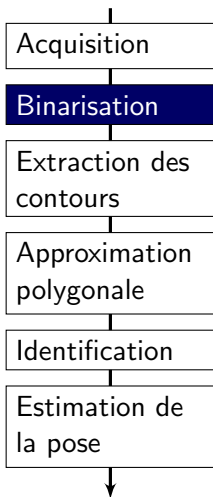
- Site/objets instrumenté ;
- Nécessite des surfaces planes ;
- Problème d'occultation des cibles

# Traitements associés à une cible codée

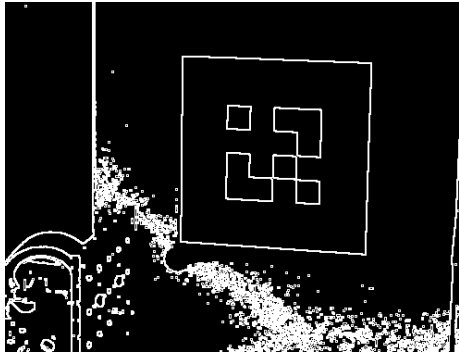
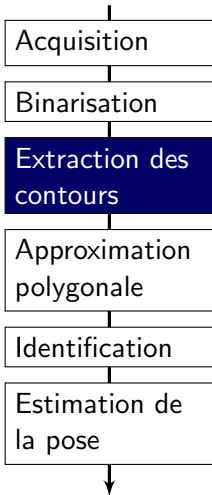




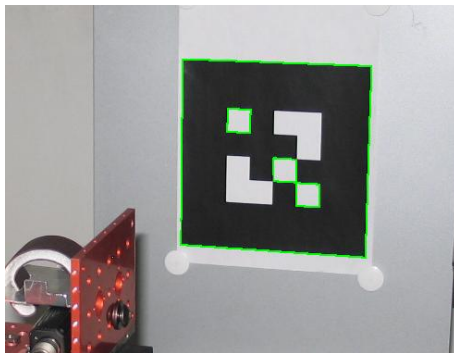
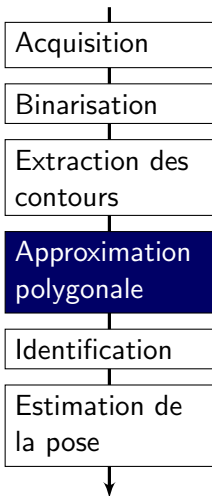
## Traitements associés à une cible codée



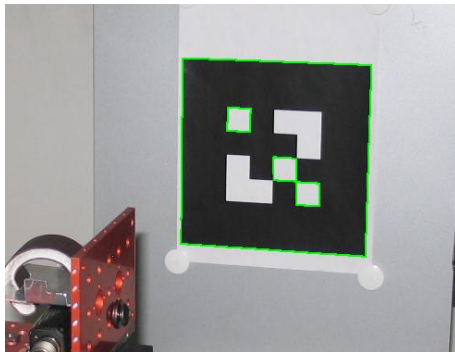
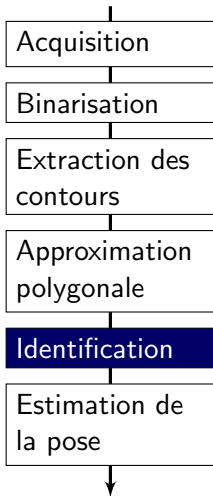
# Traitements associés à une cible codée



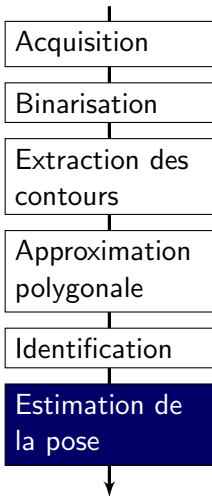
# Traitements associés à une cible codée



# Traitements associés à une cible codée



## Traitements associés à une cible codée



# Localisation basée modèle

## Primitives d'appariement

Points/sommets, descripteurs, lignes, coniques, squelettes, patches texturés, ...

## Techniques d'appariement

- Méthodes statistiques (M-estimators) ;
- RANSAC ;
- kd-tree ;
- ...

# SLAM : localisation et reconstruction simultanées

## Sources

- Plusieurs images ;
- Plusieurs caméras.

## Environnement inconnu

- Reconstruction au fur et à mesure ;
- Emploi de la géométrie épipolaire ;

# Le Projet RAXENV

## Objectifs

Démontrer l'utilisabilité d'un système de RA pour les sciences environnementales telles que la géologie.

## Nature du projet

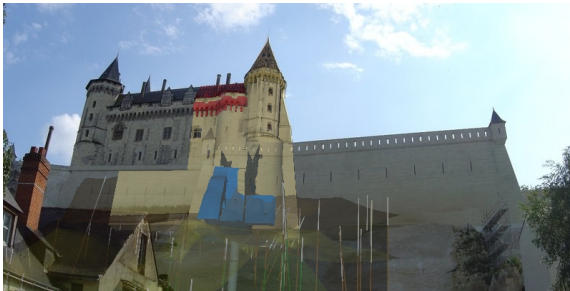
- projet exploratoire ;
- financé par l'ANR (Agence Nationale de la Recherche) ;
- démarrer en février 2007, fini en juillet 2010 ;
- consortium : BRGM, IBISC (localisation), LaBri, ArchiVideo, Lyonnaise des eaux.



# Scénario envisagé

## Château de Saumur

- RA vu en tant que moyen d'accéder à des données géolocalisées ;
- Démonstrateur utilisé en tant qu'outil d'assistance pour gérer un chantier géotechnique.



# Défis

## Défis généraux

- Localisation avec une faible latence ;
- Visualisation des données.

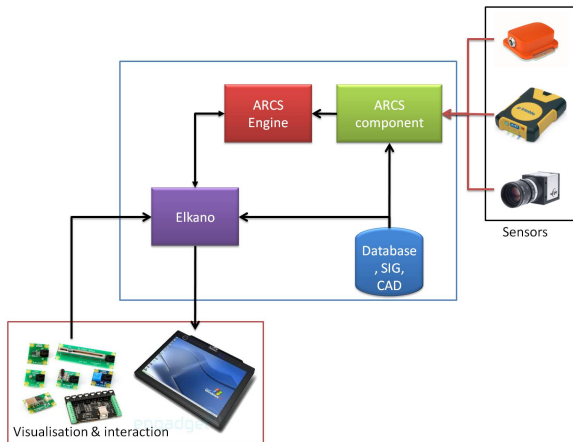
## Défis logiciels

- Intégration de plusieurs moteurs ;
- Cas concret d'utilisation du logiciel maison (ARCS – *Augmented Reality Component System*).

# Plate-forme matérielle



# Composants logiciels



# Localisation multi-capteurs

## Chaque capteur a ses défauts ...

- GPS et centrale inertielle :
  - ▶ recalibration fréquente ;
  - ▶ localisation grossière ;
- Caméra : capteur sensible aux changements de l'environnement.

## ... et ses forces

- GPS et centrale inertielle : localisation globale ;
- Camera : bonne précision.

# Localisation multi-capteurs

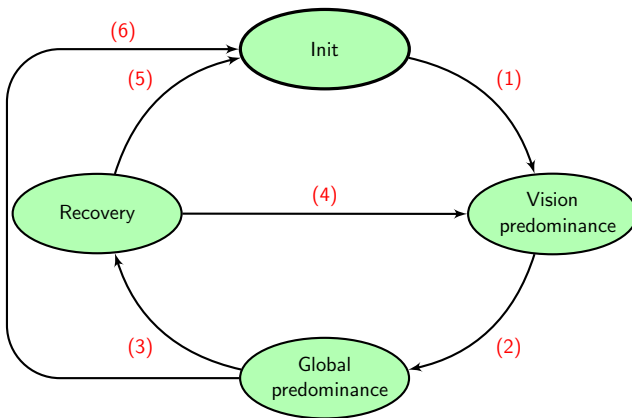
## Idée principale

Combiner les données provenant des capteurs dans l'objectif de compenser leurs faiblesses et gagner en précision et robustesse.

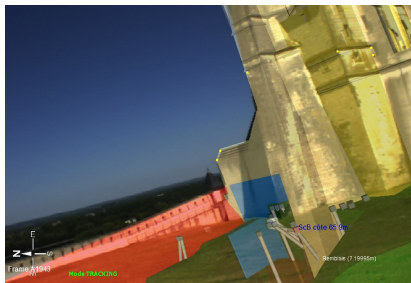
## Stratégies de combinaison

- Pratique courante : fusion de données ;
- Pratique moins courante : schéma d'assistance.

# Schéma d'assistance



# Résultats



Changements d'état

Initialisation

Visualisation in situ



## Contexte / Principe général

### RA en vision indirecte

- Une caméra filme la scène ;
- La scène augmentée contient l'image filmée par la caméra.

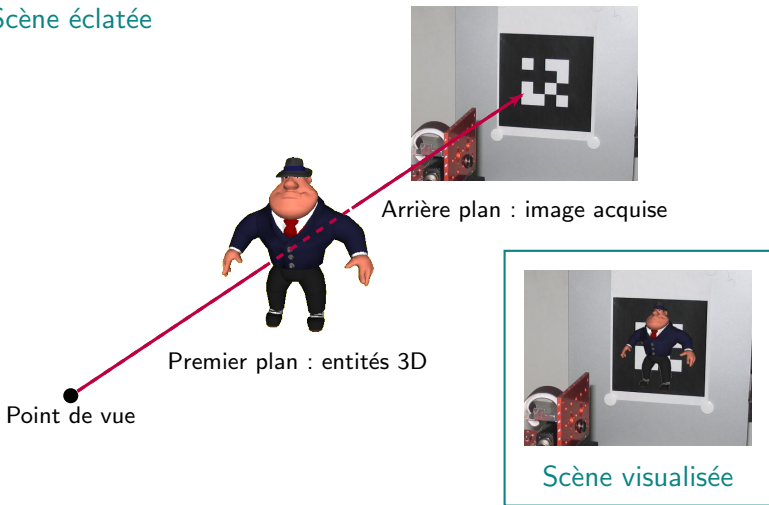
### Problématique du recalage

(Recalage : points de vue réel et virtuel alignés)

- Comment construire la scène ?
- Comment lier les caméras réelle et virtuelle ?
- Comment régler les problèmes liés aux occultations ?
- Que faire si des problèmes subsistent ?

# Composition simple

## Scène éclatée



# Caméra réelle

## Définition

- Capteur placé à un endroit et orienté d'une certaine manière dans l'espace de travail :
  - ▶ Peut-être associé à un repère local ;
  - ▶ S'il existe un repère associé à l'espace de travail, la relation liant les deux repères (orientation + position) est exprimée à l'aide des **paramètres extrinsèques**.
- Capteur créant une image 2D à partir de l'espace de travail :
  - ▶ C'est une projection modélisée par les **paramètres intrinsèques**.

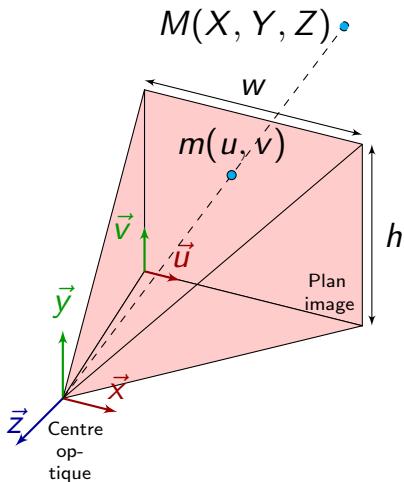
# Modèle de caméra réelle (1/2)

Rappel : modèle du sténopé (*pinhole*)

## Modèle de projection liant

- $M$ , point 3D dans l'espace ;
- $m$ , projeté de  $M$  dans l'image.

$$\begin{bmatrix} s.u \\ s.v \\ s \end{bmatrix} = \begin{bmatrix} \alpha_u & s_{uv} & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



## Modèle de caméra réelle (2/2)

Rappel : modèle du sténopé (*pinhole*)

### Paramètres (obtenus par calibration)

- $\alpha_u$  : distance focale en largeurs de pixels ;
- $\alpha_v$  : distance focale en hauteurs de pixels ;
- $u_0$  : abscisse image du projeté du point focal ;
- $v_0$  : ordonnée image du projeté du point focal ;
- $s_{uv}$  : déviation de la matrice physique (souvent négligée) ;
- $w$  : largeur en pixels de la matrice physique ;
- $h$  : hauteur en pixels de la matrice physique.

### Hypothèse de travail

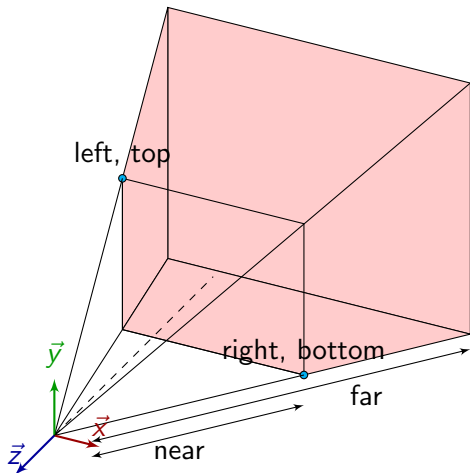
Tous les paramètres sont  $\geq 0$ .

# Modèle de caméra virtuelle

## Modèle générique

### Pyramide tronquée (*frustum*)

- Portion de l'espace considérée visible ;
- Délimitée par 6 plans de coupe (*clipping*) ;
- Espace délimité normalisé ;
- Notion de profondeur qui subsiste.



# Relation en caméras réelle et virtuelle

## Une application directe de Thales

*near* et *far* doivent être fixés :

- De manière arbitraire ;
- Ou en calculant la boîte englobante des objets de la scène.

$$left = -\frac{near}{\alpha_u} \cdot u_0$$

$$bottom = -\frac{near}{\alpha_v} \cdot v_0$$

$$right = \frac{near}{\alpha_u} \cdot (w - u_0)$$

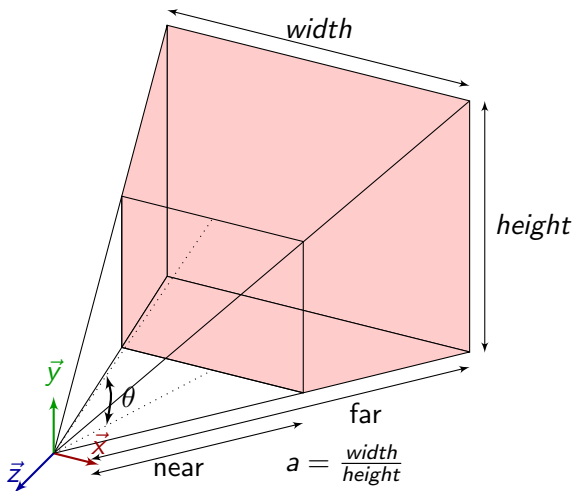
$$right = \frac{near}{\alpha_v} \cdot (h - v_0)$$

# Modèle simplifié

## Caméra perspective

Disponible pour toutes les APIs avec un jeu de paramètres simplifiés :

- $\theta$  : angle d'ouverture ;
- $a$  : rapport d'aspect ;
- $near$  et  $far$ .





## Lien caméra réelle / caméra perspective

### Idée générale

- Recouper l'image pour la rendre symétrique par rapport à la projection du centre optique ;
- *near* et *far* restent fixés.

$$a = \frac{\min(u_0, w - u_0)}{\min(v_0, h - v_0)}$$

$$\theta = 2 \cdot \text{atan} \left( \frac{\text{near}}{\alpha_v} \cdot \min(v_0, h - v_0) \right)$$

# Effectuer le rendu de l'image

Deux techniques pour l'effectuer

## Transfert pixel à pixel dans le *framebuffer*

- Pas d'anti-aliasing ;
- Pas disponible pour toutes les APIs 3D ;
- Reste très bas niveau.

## Quadrilatère texturé

- Parfois plus lent ;
- Plus complexe à mettre en oeuvre ;
- Permet de corriger les distortions radiales et tangentiellles.

## Sources de problèmes (1/2)

### Repère image (projection)

- Point d'ancrage dans le plan image ;
- Directions principales du repère image ;
- Paramètres intrinsèques positifs ou non (ex : Faugeras-Toscani).

### Repère monde (transformation spatiale)

- Placement du repère ;
- Directions (main gauche ou main droite ?) ;
- Information fournie par l'algorithme d'estimation de pose :
  - ▶ Pose de la caméra par rapport au repère monde ;
  - ▶ Pose de l'objet par rapport à la caméra.

## Sources de problèmes (2/2)

### Conventions des APIs

- Quaternions :  $w, x, y, z$  ou  $x, y, z, w$  ?
- Notations anglo-saxonnes (matrices transposées) :
  - ▶ Utilisée par les API 3D actuelles (OpenGL et Direct3D);
- Rectitude du repère :
  - ▶ OpenGL : trièdre direct (repère main droite);
  - ▶ Direct3D : trièdre indirect (repère main gauche).

### Paramètres choisis arbitrairement

Choix des paramètres *near* et *far*.

# Boîte à outils matricielle

## Inversion d'une matrice de transformation

$$M = \begin{bmatrix} R_{33} & t_{31} \\ 0_{13} & 1 \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} R_{33}^t & -R_{33}^t t_{31} \\ 0_{13} & 1 \end{bmatrix}$$

## L'explication des conventions anglo-saxonnes

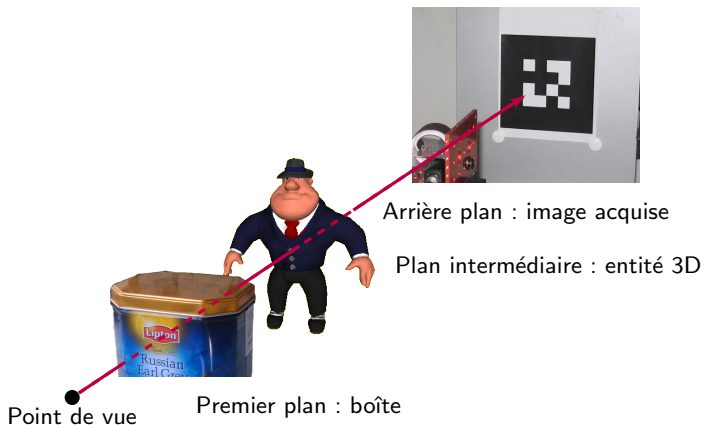
Matrice transposée = multiplication de vecteurs lignes !

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{33} & t_{31} \\ 0_{13} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad [X' \ Y' \ Z' \ 1] = [X \ Y \ Z \ 1] \begin{bmatrix} R_{33} & t_{31} \\ 0_{13} & 1 \end{bmatrix}$$

The english one

The french touch

# Scène complexe



# Gestion des occultations

## Occultations non gérées



L'augmentation perturbe la perception des profondeurs.

## Occultations gérées



L'augmentation est intégrée à son environnement.

# Apparté sur les cartes graphiques

Scène 3D = polygones + textures.

## Le travail de la carte graphique

- 1 Décomposer polygones et textures en fragments (*rasterization*) ;
  - ▶ Propriétés des fragments :
    - 1 fragment = 1 pixel dans l'image finale ;
    - Paramètres associés :  $u$ ,  $v$ , profondeur, couleur.
- 2 Traiter les fragments :
  - ▶ Les fragments passent au travers de différents tamis (*buffer*) ;
    - Les tamis sont programmables ;
    - Taille du tamis = taille de l'image finale.



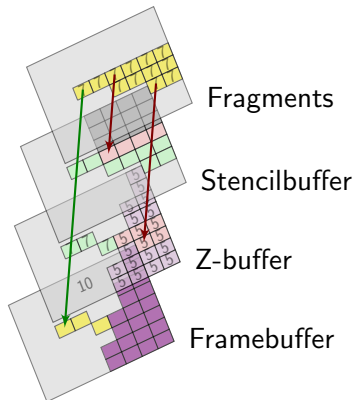
# Opérations sur les fragments

## Les tamis (*buffers*) usuels

**Framebuffer** : contient l'image finale ;

**Z-buffer** : test de profondeur ;

**Stencilbuffer** : masque ou pochoir.



# Résolution du problème des occultations

Aller plus loin que la composition simple !

La solution est à rechercher du côté de la carte graphique

## Utiliser les buffers

- **Stencilbuffer** : effectuer une première passe dedans
  - ▶ Possible si la géométrie des occultations est connue.
- **Z-buffer** : marquer directement les profondeurs :
  - ▶ Nécessite de récupérer la profondeur des objets occultants (*kinect?*).

## Faisabilité

- Cablé dans toutes les cartes graphiques actuelles ;
- Nécessite parfois un interfaçage bas niveau.

- 1 Généralités
  - Définitions
  - Chronologie, paysage et applications
- 2 Systèmes de RA
  - Partie matérielle
  - Architecture globale des traitements
  - Compétences nécessaires
- 3 Problématique et verrous
  - Localisation basée vision
  - Systèmes multi-capteurs
  - Composition des scènes de RA
    - Les difficultés
  - Les occultations réel/virtuel
- 4 Architectures logicielles

# Besoins en architecture

## Objectif

- Créer rapidement des applications de RA ;
- Appliquer une méthode à un contexte.

## Contraintes

- Discipline jeune (les méthodes évoluent) ;
- Capteurs expérimentaux.

## Moyens

- Réutiliser rapidement le code déjà développé ;
- Standardiser le code pour le remplacer facilement.

# Les solutions récurrentes

## Problème et solution récurrente

- Une trentaine de projets depuis une dizaine d'années !  
[Endres2005]
- Un schéma récurrent : l'utilisation de composants.

## Définition : composant

Morceau de logiciel (script, service ou code compilé) sujet à composition avec d'autres composants en vue de former un logiciel complet.

# Types d'approches

## Approche flux de données (*dataflow*)

- Communauté de traitement de l'image ;
- Accent mis sur le cheminement des données entre divers éléments de traitement.

## Approche graphe de scène (*scenegraph*)

- Communauté de réalité virtuelle ;
- Accent mis sur les hiérarchie des données.

## Des approches complémentaires, non pas antagonistes

Les deux cohabitent souvent, simplement un des deux aspects est privilégié.

# Bibliothèques et logiciels pour la RA (1/2)

## Bibliothèque de cibles codées

- ARToolkit : <http://www.hitl.washington.edu/artoolkit/>
- ARTag : <http://www.artag.net/>
- ARUCO : <http://www.uco.es/investiga/grupos/ava/node/26>

## Frameworks de laboratoires de recherche

- DWARF : <http://ar.in.tum.de/Chair/ProjectDwarf>
- Studierstube : <http://studierstube.icg.tugraz.at/main.php>
- Tinmith : <http://www.tinmith.net/>
- ARCS : <http://arcs.ibisc.univ-evry.fr> (non public)

## Bibliothèques et logiciels pour la RA (2/2)

### Exemple de briques plus fondamentales (intégration)

- Traitement d'images : OpenCV ;
- Gestion de la 3D : Ogre, Coin3D, OpenSceneGraph ;
- Gestion de la distribution : CORBA ;
- Interface graphique : Qt ;
- Gestion des périphérique de RV : VRPN.

**Et bien sûr les solutions logicielles des entreprises citées !**