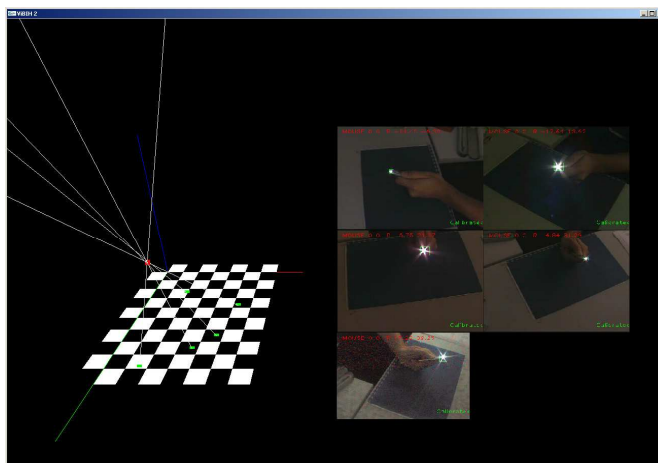


Version du	30/11/10	<b>Secrétariat général HES-SO</b>	DT03
Créé par	JSE		
		<b>ViBIH 2</b>	
		<b>Rapport suivi 3D</b>	

## Tracking multi-camera en 3D



Julien Senn  
Philippe Etique  
François Tièche

3.12.2010

## **Table des matières**

<b>1. Introduction.....</b>	<b>3</b>
<b>2. Calibrage.....</b>	<b>3</b>
<b>3. Position d'un point suivi par plusieurs caméras.....</b>	<b>4</b>
3.1 Paramètres extrinsèques.....	4
3.2 Calcul de la droite passant par le centre optique des caméras et l'objet ponctuel suivi.....	5
<b>4. Calibrage multi-mire .....</b>	<b>6</b>
<b>5. Implémentation .....</b>	<b>10</b>
5.1 Architecture.....	10
5.2 Utilisation.....	11
5.2.1 Calibrage intrinsèque .....	11
5.2.2 Calibrage extrinsèque.....	12
5.2.3 Calibrage extrinsèque simultané .....	12
5.2.4 Calibrage extrinsèque itératif.....	12
5.2.5 Tracking sur l'interface.....	12
5.3 Mode d'emploi .....	13
<b>6. Expériences.....</b>	<b>14</b>
6.1 Principe de mesure.....	14
6.2 Expérience.....	15
6.3 Résultat.....	16
<b>7. Conclusion .....</b>	<b>16</b>
<b>8. ANNEXE 1.....</b>	<b>17</b>

## 1. Introduction

Ce rapport expose la manière dont est déterminée la position tridimensionnelle d'un objet suivi par un tracker multi-caméra. Les buts de ce tracker sont de suivre un objet dans l'espace à l'aide de plusieurs caméras calibrées, de représenter sa trajectoire en 3 dimensions et finalement d'analyser cette dernière.

Tout d'abord une introduction présente le calibrage d'une caméra. Ensuite, la méthode de calibrage de plusieurs caméras est expliquée et la manière de déterminer la position de l'objet suivi est montrée. Le paragraphe « Calibrage multi-mire » décrit comment un système, où les caméras peuvent être positionnées n'importe où dans l'environnement, est calibré. Ensuite, les expériences menées et les mesures effectuées sont présentées. Enfin, les principales conclusions sont discutées.

## 2. Calibrage

Le calibrage d'une caméra est la tâche qui consiste à trouver les paramètres qui permettent d'exprimer les coordonnées d'un point, défini dans un repère 3D lié à l'environnement, dans un autre repère, lié au plan image de la caméra. Ces paramètres sont regroupés dans des matrices dites intrinsèques et extrinsèques.

Soit un point P (X, Y, Z) exprimé dans le repère lié à l'environnement. Les coordonnées homogènes  $(\omega \cdot u, \omega \cdot v, \omega)^t$  de ce même point, exprimées dans le repère lié au plan image sont :

$$\begin{bmatrix} \omega \cdot u \\ \omega \cdot v \\ \omega \end{bmatrix} = \underbrace{\begin{bmatrix} k_u & s_{uv} & c_u \\ 0 & k_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Paramètres intrinsèques}} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} & & & t_x \\ & R_{3 \times 3} & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Paramètres extrinsèques}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.1)$$

Paramètres intrinsèques (propre à la caméra) :

- $f$  distance focale en pixels.
- $k_u$  et  $k_v$  facteurs d'agrandissement de l'image.
- $c_u$  et  $c_v$  coordonnées en pixels de la projection du centre optique de la caméra sur le plan image.
- $s_{uv}$  traduit la non-orthogonalité potentielle des lignes et des colonnes de cellules électroniques photosensibles qui composent le capteur de la caméra.

Paramètres extrinsèques (variant selon la position de la caméra) :

- $R_{3 \times 3}$  matrice de rotation permettant de passer du repère lié à l'environnement au repère lié au centre optique de la caméra.
- $t_x, t_y$  et  $t_z$  composantes, en mètres, du vecteur de translation permettant de passer du repère lié à l'environnement au repère lié au centre optique de la caméra.

La librairie OpenCV propose un algorithme [Bouguet] qui permet de calculer les paramètres intrinsèques et extrinsèques à partir d'images d'une mire en forme d'un damier noir et blanc. Pour obtenir des paramètres intrinsèques de bonne qualité, la mire doit être présentée à la caméra un grand nombre de fois dans des positions différentes. Pour effectuer le calibrage extrinsèque, OpenCV fixe le repère lié à l'environnement en haut à gauche du damier (figure 10). Ce calibrage ne nécessite qu'une seule vue de la mire.

### 3. Position d'un point suivi par plusieurs caméras

La méthode de calcul de la position tridimensionnelle d'un objet ponctuel suivi par plusieurs caméras est présentée ci-dessous. Le système de coordonnées de référence coïncide avec celui d'une mire de référence posée au sol. Toutes les caméras sont calibrées par rapport à cette mire. Il est ainsi possible de déterminer la position du centre optique de toutes les caméras par rapport à ce référentiel. Connaissant la position de l'objet à suivre dans le plan image de chacune des caméras, il est possible de déterminer, pour chaque caméra, une droite passant par son centre optique et par le point du plan image correspondant à l'objet suivi. La position de l'objet est l'intersection de toutes les droites.

#### 3.1 Paramètres extrinsèques.

Dans un premier temps, l'hypothèse que toutes les caméras voient en entier la mire posée au sol est faite. Cette hypothèse simplifie la phase de calibrage. La méthode de Bouguet définit automatiquement un référentiel lié à cette mire de calibrage (figure 1), qui sera considéré comme le référentiel de base, lié au monde  $O^W$ . Un cas général est montré plus loin.

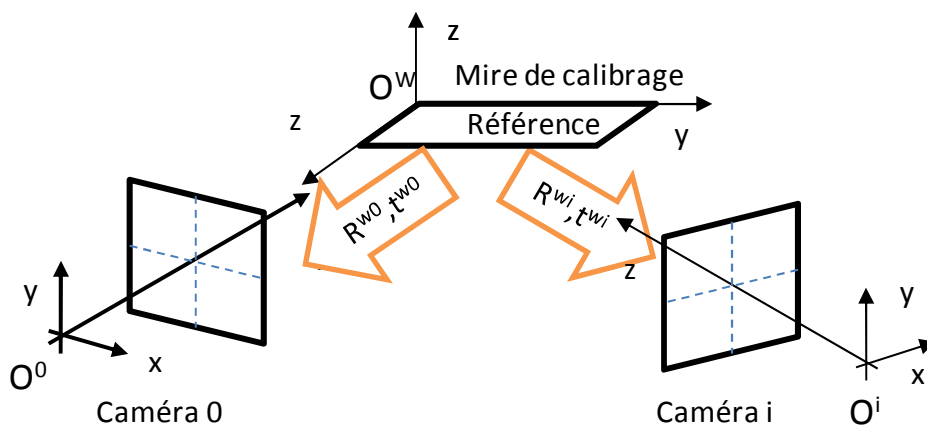


Figure 1 : Système de référence lié au monde  $O^W$  et système de coordonnées liés aux centres optiques des caméras  $O^i$ . Les paramètres de calibrage extrinsèques sont en orange

Les caméras sont calibrées par rapport à cette mire, et pour chaque caméra  $i$ , la phase de calibrage extrinsèque fournit une matrice de rotation  $R^{wi}$  et un vecteur de translation  $t^{wi}$ .

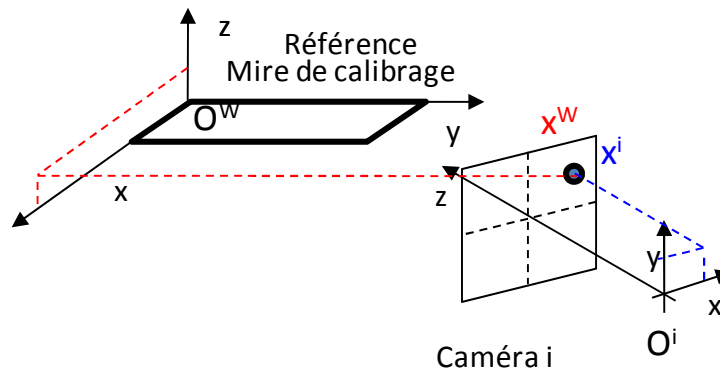


Figure 2 : Point du plan image, exprimé dans le repère lié au monde et dans le repère lié au centre optique.

Connaissant les coordonnées d'un point dans le référentiel du monde  $O^W$ , il est alors possible de déterminer ses coordonnées dans le référentiel lié au centre optique  $O$  de la  $i^{\text{ème}}$  caméra.

$$\vec{x}^i = R^{Wi} \cdot \vec{x}^W + \vec{t}^{Wi} \tag{3.1}$$

L'opération inverse est :

$$\vec{x}^W = (R^{Wi})^{-1} \cdot (\vec{x}^i - \vec{t}^{Wi}) = (R^{Wi})^T \cdot (\vec{x}^i - \vec{t}^{Wi}) \tag{3.2}$$

### 3.2 Calcul de la droite passant par le centre optique des caméras et l'objet ponctuel suivi.

Le centre optique  $c^{Wi}$  de la  $i^{\text{ème}}$  caméra dans le repère lié au monde est déterminé en exprimant l'origine du système caméra,  $x^i=(0,0,0)^T$ , dans le repère du monde :

$$\vec{c}^{Wi} = (R^{Wi})^T \cdot (-\vec{t}^{Wi}) \tag{3.3}$$

La position de l'objet ponctuel suivi, vu par une caméra, est définie par ses coordonnées écran. Ses coordonnées dans le repère lié au centre optique de la caméra sont déterminées à l'aide des paramètres intrinsèques de la caméra. Pratiquement, il n'est pas possible de déterminer la position 3D de l'objet, mais uniquement l'ensemble de points correspondant à la droite sur laquelle se trouve l'objet.

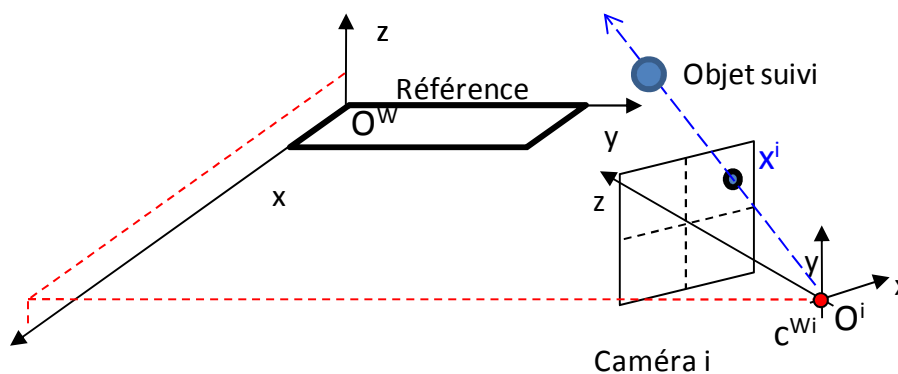


Figure 3 : Position du centre optique  $c^{Wi}$  de la caméra  $i$ , exprimé dans le repère monde et la droite indiquant la direction de l'objet suivi

Connaissant les coordonnées du centre optique de la caméra et du point image sur l'écran, il est dès lors possible d'exprimer, dans le repère lié au monde, l'équation de la droite passant par le centre optique de chaque caméra et l'objet suivi.

La position de l'objet est alors l'intersection de toutes ces droites (figure 4). En pratique, ce point n'existe pas car les droites ne se croisent pas. Il s'agit donc de trouver un point situé à proximité de toutes les droites.

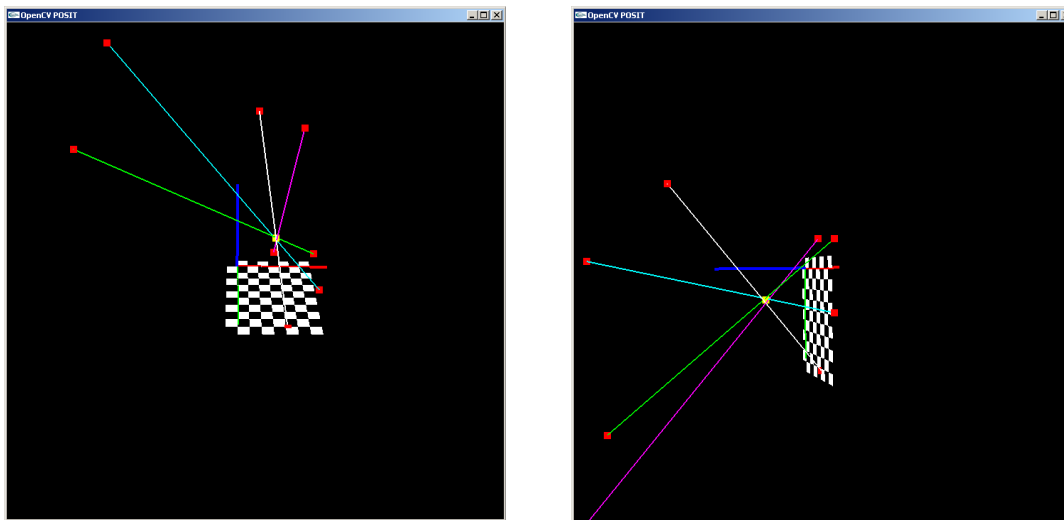


Figure 4 : Intersection des droites issues des caméras

L'algorithme choisi consiste à prendre successivement toutes les droites 2 par 2, ce qui fait  $n(n-1)/2$  opérations pour  $n$  droites. Il s'agit, pour chaque couple de droites, de calculer le segment le plus court entre elles (voir Annexe 1) et à prendre le point milieu de ce segment. Finalement, les coordonnées tridimensionnelles de l'objet filmé sont déterminées en moyennant l'ensemble de tous les points milieux.

#### 4. Calibrage multi-mire

Dans le paragraphe précédent, l'hypothèse que toutes les caméras voyaient directement la mire de référence a été faite et pouvaient donc s'y calibrer. En réalité, une telle situation est improbable car les caméras peuvent être placées n'importe où dans une pièce. De plus, il n'est pas possible de disposer d'une mire couvrant la totalité du sol dans des pièces habitables.

Ci-dessous, une méthode de calibrage se basant sur un ensemble de caméras réparti de manière quelconque dans l'environnement est décrite. Une mire posée au sol constitue la référence du monde. Le principe adopté est de calibrer au moins une caméra par rapport à la mire de référence, puis de calibrer les autres caméras de proche en proche en présentant une mire visible par au moins deux caméras.

Un point  $\vec{x}^i$  peut être lié au référentiel du monde  $\vec{x}^W$  dans le repère lié à la caméra  $i$  :

$$\vec{x}^i = R^{Wi} \cdot \vec{x}^W + \vec{t}^{Wi} \tag{4.1}$$

A l'inverse, un point  $\vec{x}^i$  peut aussi être lié au référentiel de la caméra  $i$  dans le référentiel lié au monde  $\vec{x}^W$  :

$$\vec{x}^W = (R^{Wi})^{-1} \cdot (\vec{x}^i - \vec{t}^{Wi}) = (R^{Wi})^T \cdot (\vec{x}^i - \vec{t}^{Wi}) \tag{4.2}$$

$$\begin{aligned} \vec{x}^W &= R^{iW} \vec{x}^i + \vec{t}^{iW} \\ \text{avec } R^{iW} &= (R^{Wi})^T \\ \text{et } \vec{t}^{iW} &= -(R^{Wi})^T \vec{t}^{Wi} \end{aligned} \tag{4.3}$$

Le principe adopté (figure 5) est de considérer une caméra calibrée par rapport à la mire de référence. Par exemple si les paramètres intrinsèques  $R^{Wi}$  et  $t^{Wi}$  de la  $i^{\text{ème}}$  caméra sont connus, les paramètres de la caméra  $j$  sont à déterminer, lorsque cette dernière ne voit pas la mire de référence, mais uniquement la  $p^{\text{ème}}$  mire, également visible par la caméra  $i$ . Les paramètres de calibrage extrinsèque de ces deux caméras par rapport à la mire  $p$  sont :  $R^{pi}, t^{pi}$  et  $R^{pj}, t^{pj}$ . Le problème consiste à déterminer les paramètres de la caméra  $j$  par rapport à la mire de référence.

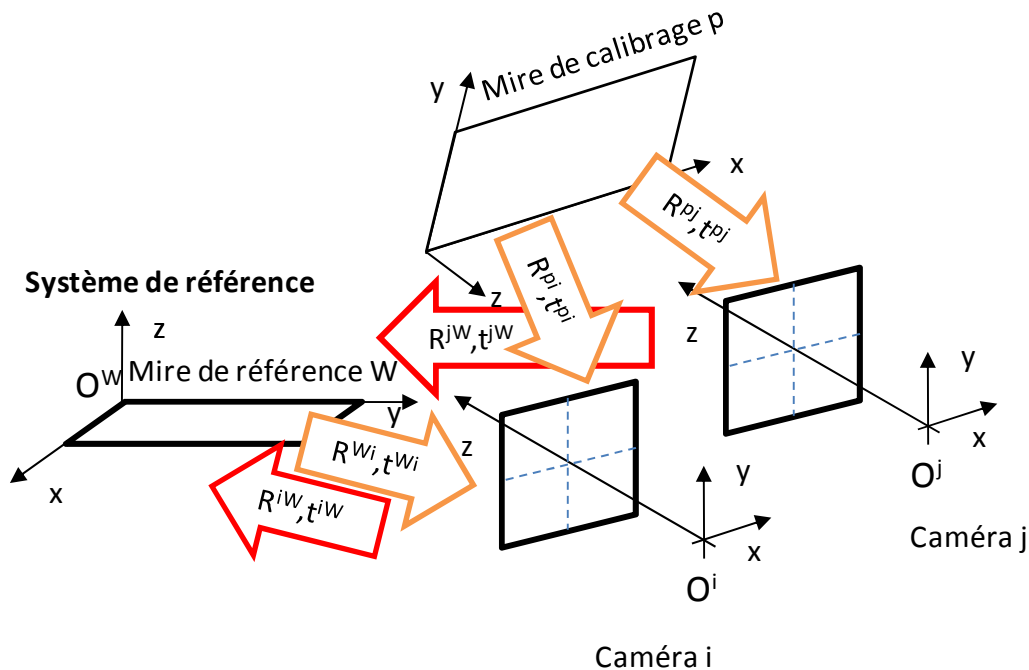


Figure 5 : Calibrage multi-mire

**But :** Déterminer la matrice et la translation permettant d'exprimer un point lié au référentiel de la caméra  $j$  dans le repère du monde.

$$\vec{x}^W = R^{jW} \cdot \vec{x}^j + t^{jW} \tag{4.4}$$

**Hypothèse :**

- a) Soit une caméra  $i$  dont les paramètres de changement de repère avec le référentiel d'origine sont connus (4.3),

$$\vec{x}^W = R^{iW} \cdot \vec{x}^i + t^{iW} \tag{4.5}$$

- b) La caméra  $i$  a été calibrée conjointement avec la caméra  $j$  sur la mire  $p$ . Ainsi les transformations sont obtenues  $R^{pi}, t^{pi}, R^{pj}$  et  $t^{pj}$  avec les relations suivantes:

$$\vec{x}^i = R^{pi} \cdot \vec{x}^p + t^{pi} \tag{4.6}$$

$$\vec{x}^j = R^{pj} \cdot \vec{x}^p + t^{pj} \tag{4.7}$$

De l'équation (4.7)  $x^p$  peut être exprimé par:

$$\vec{x}^p = (R^{pj})^T (\vec{x}^j - t^{pj}) \tag{4.8}$$

Remarque : Dans les équations (4.4) et (4.5)  $R^{iW}$  et  $\vec{t}^{iW}$  ne sont pas les matrices issues du calibrage, mais des matrices « inverses », voir (4.3). Cette notation a été préférée, car les changements de coordonnées vont toujours du référentiel-caméra au référentiel-monde.

1) Pas initial de calibrage de la caméra 1, par rapport à la mire de référence posée au sol.

Soient  $R^{W1}$  et  $\vec{t}^{W1}$  la matrice de rotation et le vecteur de translation obtenus par calibrage extrinsèque de la caméra 1, par rapport à la mire de référence W. Ceci permet d'exprimer un point dans le référentiel de la caméra 1 à partir de ses coordonnées liées au monde:

$$\vec{x}^1 = R^{W1} \cdot \vec{x}^W + \vec{t}^{W1} \quad (4.9)$$

Dans le cas contraire, on exprime un point lié au monde à partir de ses coordonnées dans le système de la caméra 1 :

$$\vec{x}^W = (R^{W1})^T \cdot (\vec{x}^1 - \vec{t}^{W1}) \quad (4.9)$$

$$\vec{x}^W = \underbrace{(R^{W1})^T}_{R^{1W}} \cdot \vec{x}^1 - \underbrace{(R^{W1})^T}_{\vec{t}^{1W}} \vec{t}^{W1} \quad (4.10)$$

Ainsi, un point en coordonnées monde est exprimé en fonction de coordonnées caméra:

$$\vec{x}^W = R^{1W} \cdot \vec{x}^1 + \vec{t}^{1W}$$

avec

$$R^{1W} = (R^{W1})^T \quad (4.11)$$

$$\vec{t}^{1W} = -(R^{W1})^T \vec{t}^{W1}$$

2) Au pas p, calibrage par rapport à une mire mobile, le calibrage des la caméra  $i$  est supposé connu et le calibrage de la caméra  $j$  est cherché.

En remplaçant la valeur de  $\vec{x}^i$  de l'équation (4.6) dans l'équation (4.11), on obtient:

$$\vec{x}^W = R^{iW} \cdot \vec{x}^i + \vec{t}^{iW} \stackrel{\vec{x}^i = R^{pi} \cdot \vec{x}^p + \vec{t}^{pi}}{=} R^{iW} \cdot (R^{pi} \cdot \vec{x}^p + \vec{t}^{pi}) + \vec{t}^{iW} \quad (4.12)$$

Dans (4.12),  $\vec{x}^p$  est remplacée par sa valeur dans (4.8):

$$\begin{aligned} \vec{x}^W &= R^{iW} \cdot (R^{pi} \cdot \vec{x}^p + \vec{t}^{pi}) + \vec{t}^{iW} \stackrel{\vec{x}^p = (R^{pj})^T (\vec{x}^j - \vec{t}^{pj})}{=} \\ &= R^{iW} \cdot \left( R^{pi} \cdot \left( (R^{pj})^T (\vec{x}^j - \vec{t}^{pj}) \right) + \vec{t}^{pi} \right) + \vec{t}^{iW} \end{aligned} \quad (4.13)$$

En développant:

$$\vec{x}^W = \underbrace{\left( R^{iW} \cdot R^{pi} \cdot (R^{pj})^T \right)}_{R^{jW}} \vec{x}^j + \underbrace{\vec{t}^{iW} + R^{iW} \cdot \vec{t}^{pi} - R^{iW} \cdot R^{pi} \cdot (R^{pj})^T \cdot \vec{t}^{pj}}_{\vec{t}^{jW}} \quad (4.14)$$

Ainsi :

$$R^{jW} = R^{iW} \cdot R^{pi} \cdot (R^{pj})^T \quad (4.15)$$



$$t^{jW} = t^{iW} + R^{iW} \cdot t^{pi} - R^{iW} \cdot R^{pi} \cdot (R^{pj})^T \cdot t^{pj} \quad (4.16)$$

## Résumé

Connaissant les matrices et translations permettant le changement de coordonnées

de la i<sup>ème</sup> caméra vers le monde :  $R^{iW}, t^{iW}$  par  $\vec{x}^W = R^{iW} \cdot \vec{x}^i + \vec{t}^{iW}$ ,

de la p<sup>ème</sup> mire vers la i<sup>ème</sup> caméra :  $R^{pi}, t^{pi}$  par  $\vec{x}^i = R^{pi} \cdot \vec{x}^p + \vec{t}^{pi}$ ,

de la p<sup>ème</sup> mire vers la j<sup>ème</sup> caméra :  $R^{pj}, t^{pj}$  par  $\vec{x}^j = R^{pj} \cdot \vec{x}^p + \vec{t}^{pj}$ ,

les paramètres extrinsèques de la j<sup>ème</sup> caméra vers le monde sont

$$R^{jW} = R^{iW} \cdot R^{pi} \cdot (R^{pj})^T \quad (4.17)$$

$$t^{jW} = t^{iW} + R^{iW} \cdot t^{pi} - R^{iW} \cdot R^{pi} \cdot (R^{pj})^T \cdot t^{pj} \quad (4.18)$$

Cela permet d'exprimer un point de la caméra j dans le repère du monde.

$$\vec{x}^W = R^{jW} \cdot \vec{x}^j + \vec{t}^{jW} \quad (4.19)$$

## 5. Implémentation

L'application de tracking en 3 dimensions a été développée en C/C++ sur une plateforme Windows. Il s'agit d'une application multithread qui comporte un processus propre pour chaque caméra ainsi qu'un processus de fusion des données et d'affichage. La plupart des outils de calibrage ont été fournis par la librairie de traitement d'image OpenCV et l'interface utilisateur utilise OpenGL pour l'affichage 3D.

### 5.1 Architecture

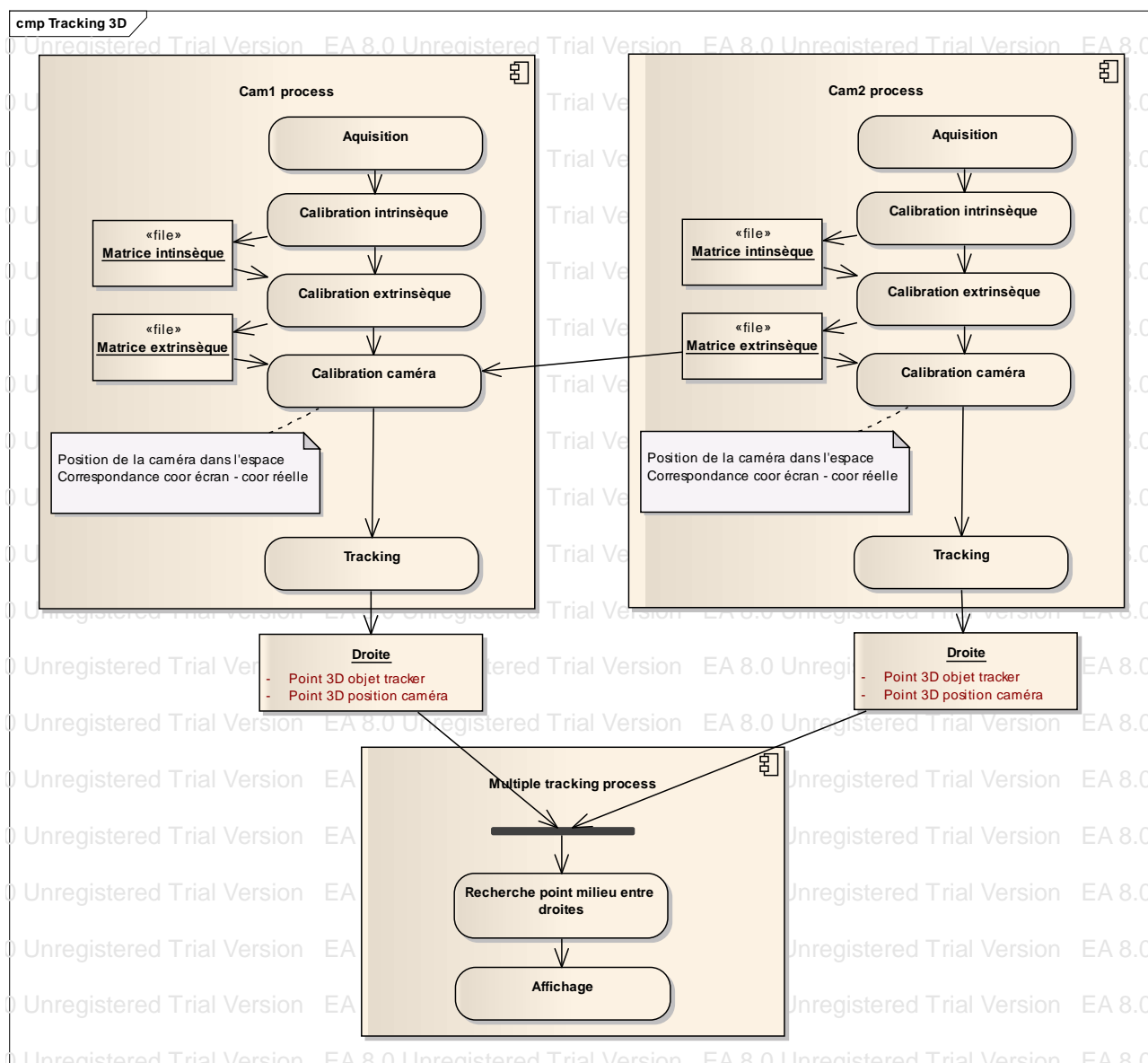


Figure 6 : Architecture du programme

## 5.2 Utilisation

Plusieurs fonctionnalités ont été développées :

- Calibrage intrinsèque
- Calibrage extrinsèque
- Calibrage extrinsèque simultanée
- Calibrage par l'intermédiaire d'une caméra déjà calibrée
- Tracking sur l'interface

### 5.2.1 Calibrage intrinsèque

Lors de sa première utilisation, le logiciel recherche les paramètres intrinsèques propres à chaque caméra. Ces derniers restent les mêmes tout au long de l'expérience. Pour ce faire, un damier est présenté à plusieurs reprises à chaque caméra selon différentes orientations et distances dans le but de déterminer leurs caractéristiques propres (focale,...). Cela est réalisé à l'aide de la méthode `cvCalibrateCamera2` de la librairie de traitement d'image OpenCV. La phase de calibrage intrinsèque doit être faite lors de la première utilisation du programme car un fichier est généré lors de cette opération. Par la suite le logiciel utilise ce fichier.

La mire de référence utilisée est un damier de 10 carrés sur 7 et chaque carré mesure 23 mm de côté. Ce damier est présenté 50 fois, successivement, dans différentes positions à la caméra.

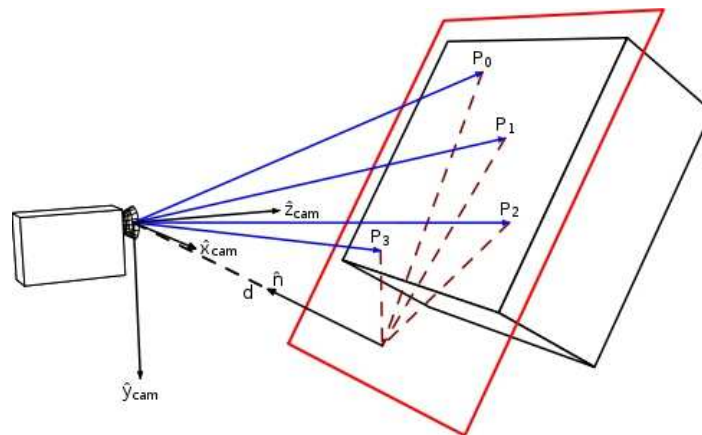


Figure 7 : Prises de vue de la mire à différents angles et distances

### 5.2.2 Calibrage extrinsèque

---

Après avoir déterminé les paramètres intrinsèques de chaque caméra, le système complet est installé (figure 8). Les caméras ne devront plus être déplacées à partir du moment où les paramètres extrinsèques seront fixés.



Figure 8 : Exemple de montage

Lors du montage, toutes les caméras doivent être placées de manière à voir la mire en entier.

Pour générer la matrice extrinsèque d'une caméra, la mire étant visible en entier, il suffit d'exécuter l'algorithme de détection et d'analyse du damier. Une fois la mire détectée, la méthode d'OpenCV `cvFindExtrinsicCameraParams2` génère la matrice extrinsèque correspondante.

### 5.2.3 Calibrage extrinsèque simultané

---

Le mode simultané de recherche de la matrice extrinsèque nécessite que la mire soit visible par toutes les caméras au même moment. Ainsi l'algorithme génère une matrice intrinsèque pour chaque caméra.

### 5.2.4 Calibrage extrinsèque itératif

---

Lorsque la mire n'est pas visible de toutes les caméras, il est aussi possible de calibrer le système complet (voir le chapitre Calibrage Multi-mire). Pour cela on utilise le fait qu'une caméra soit déjà calibrée pour calibrer la suivante en plaçant la mire de manière à ce qu'elle soit visible en entier aussi bien par la première caméra que par la seconde. La nouvelle matrice ainsi obtenue pour la première caméra est envoyée à la seconde qui la transforme pour en changer le référentiel.

### 5.2.5 Tracking sur l'interface

---

Une fois le système configuré, un objet peut être suivi. Dans l'exemple présenté ci-dessous, c'est une source lumineuse (lampe de poche) qui est suivie.

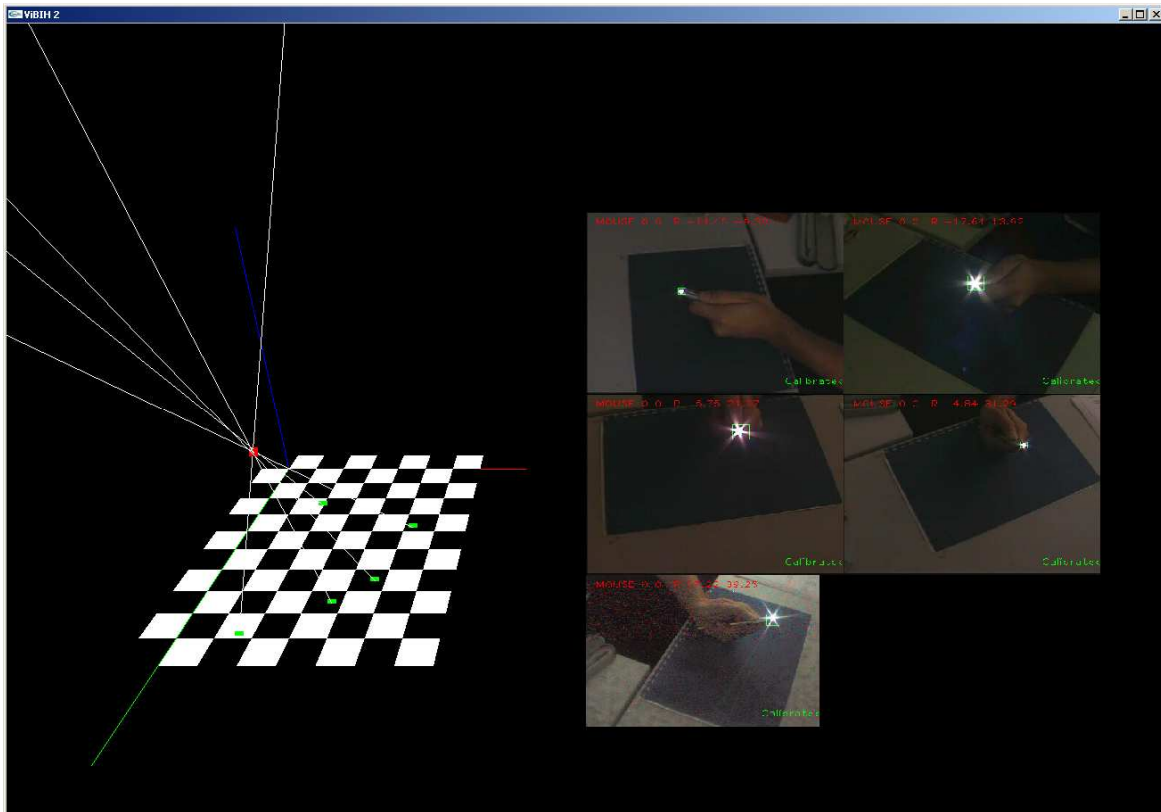


Figure 9 : Exemple de suivi d'une source lumineuse

La trajectoire est reportée sur la représentation 3D de l'environnement (partie gauche de l'interface). Un composant OpenGL permet d'observer la scène sur 360°.

### 5.3 Mode d'emploi

Des touches de raccourcis permettent d'activer différentes fonctions dans l'interface :

Dans la partie interface principale :

- [s] En plus de la fenêtre d'interface, affiche une fenêtre propre à chaque caméra. Cela permet de calibrer chacune des caméras indépendamment.
- [c] Lance l'algorithme de génération simultanée des matrices extrinsèques pour toutes les caméras.

Sur les fenêtres propres aux caméras :

- [i] Lance l'algorithme de génération de la matrice intrinsèque.
- [e] Lance l'algorithme de génération de la matrice extrinsèque.
- [w] Pour une caméra déjà calibrée, écrit le fichier contenant la matrice extrinsèque lors d'une calibration par intermédiaire.
- [r] Pour une caméra pas encore calibrée, lit le fichier contenant la matrice extrinsèque d'une caméra déjà calibrée lors d'un calibrage itératif.

## 6. Expériences

Les expériences décrites ci-dessous utilisent le même système que celui qui a été mis en place pour l'application de tracking 3D (figure 8).

### 6.1 Principe de mesure

Différents tests ont permis de mesurer la précision du système :

Une fois la phase de calibrage terminée, les mesures ont été effectuées en plaçant un objet de référence, dont la géométrie est connue, devant les caméras. Les résultats calculés par le système sont comparés à la géométrie de l'objet. L'objet de référence est constitué de damiers de différentes tailles placés à différents niveaux. Les coordonnées des points caractéristiques des damiers sont déterminées et comparées avec leurs positions réelles.

La taille des damiers a été adaptée pour permettre à toutes les caméras de les voir en entier malgré la hauteur de l'objet (figure 10 et 11).

- Hauteur de 0 mm : Damier 0 de 10 carrés de long, 7 de large, carrés de 23 mm de côté,
- Hauteur de 35 mm : Damier 1 de 7 carrés de long, 6 de large, carrés de 23 mm de côté,
- Hauteur de 45 mm : Damier 2 de 6 carrés de long, 5 de large, carrés de 23 mm de côté

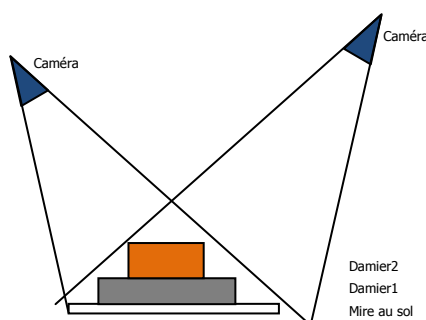


Figure 10: Plan de coupe du montage

Mire au sol	Damier1	Damier2
taille 10 * 7, haut 0 mm	taille 7 * 6, haut 35 mm	taille 6 * 5, haut 45 mm

Figure 11: Objet de référence

## 6.2 Expérience

Trois expériences ont été effectuées dans le but d'évaluer la précision du calibrage multi-caméra proposé. Les caméras sont placées dans les mêmes positions dans les trois expériences.

La première expérience (A) consiste à calibrer directement chaque caméra par rapport à la mire de référence. La deuxième expérience (B), consiste à calibrer uniquement la première caméra par rapport à la grille, puis à calibrer de proche en proche les caméras en leurs présentant une mire placée dans l'espace (Voir Calibrage Multi-mire). La troisième expérience (C) consiste à appliquer le principe de calibrage utilisé dans l'expérience B, mais en se ramenant à la situation où toutes les caméras voient simultanément la mire. Celle n'a alors pas besoin d'être déplacée. Cette expérience permet de montrer une étape intermédiaire entre les étapes A et B.

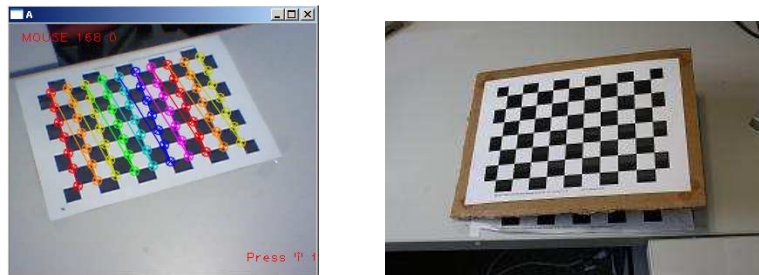


Figure 12: Calibrage sur la mire de référence et mire de calibrage mobile

Une fois le système calibré, le damier1 est placé sur la mire au sol le damier1 (figure 13). Le logiciel recherche tous les points qui le composent et calcule leurs coordonnées dans l'espace. Les points trouvés sont ensuite comparés aux points réels.



Figure 13: Recherche des points du damier1

Ensuite, l'opération est effectuée avec le damier2 et enfin, en superposant le damier2 sur le damier1.

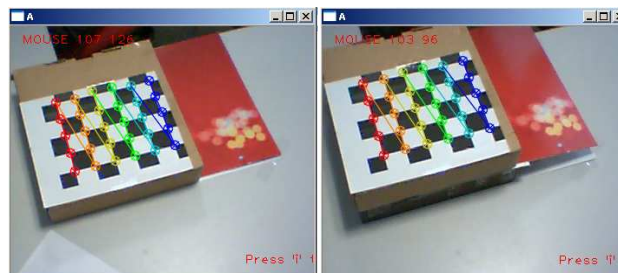


Figure 14: Recherche des points du damier2 et superposition

L'écart entre les points réels et les points détectés permet d'évaluer la précision.

**6.3 Résultat**

	<b>Ecart (cm)</b>	<i>Damier1</i> $\Delta z = 35mm$	<i>Damier2</i> $\Delta z = 45mm$	<i>Superposition</i> $\Delta z = 80mm$
Expérience A	<i>moyen</i>	0.10	0.20	0.20
	<i>max</i>	0.10	0.20	0.20
	<i>RMS</i>	<b>0.16</b>	<b>0.25</b>	<b>0.29</b>
	<i>RMS</i>	<b>0.23</b>		
Expérience B	<i>moyen</i>	0.4	0.7	0.6
	<i>max</i>	0.8	0.9	<b>1.0</b>
	<i>RMS</i>	<b>0.524</b>	<b>0.68</b>	<b>0.75</b>
	<i>RMS</i>	<b>0.64</b>		
Expérience C	<i>moyen</i>	0.1	0.3	0.2
	<i>max</i>	0.3	0.3	0.3
	<i>RMS</i>	<b>0.15</b>	<b>0.25</b>	<b>0.32</b>
	<i>RMS</i>	<b>0.24</b>		

L'imprécision des mesures n'augmente que peu lorsque la coordonnée z augmente.

Une faible différence (0.23cm et 0.24cm) entre les expériences A et C est observée. Comme la mire de calibrage est toujours la même, les erreurs constatées sont principalement dues aux opérations de multiplications de matrices nécessitées par la méthode de calibrage multi-cible, ainsi qu'aux imprécisions humaine (placement des damiers, vibrations du système).

La différence (0.23cm et 0.64cm) entre les expériences A et B, est surtout induite par des erreurs de détection des points. En effet, sous certains angles d'observation de la mire, les points sont plus difficilement détectés. Comme pour le cas de la différence entre A et C, elle est aussi due aux opérations de multiplications de matrices.

Mais dans le pire des cas, et à 8 cm du sol, une erreur 1 cm est détectée dans l'expérience B. Une telle valeur peut être négligée dans le cas du suivi d'une trajectoire, mais qui peut poser problème lorsqu'une métrique précise est exigée.

**7. Conclusion**

Une méthode de calibrage multi-caméra utilisée pour faire du tracking tridimensionnelle d'objets ponctuel a été développée. Cette solution peut être intégrée dans un système à taille humaine où l'objet suivi est la tête d'une personne et dont la trajectoire, une fois étudiée, permet de détecter une chute.

Le calibrage basé sur une mire fixe donne les meilleurs résultats, mais dans un cas réel, il est difficile de placer toutes les caméras de façon à leurs permettre de voir la même mire.

Le système montre de bonnes performances avec une erreur maximale de l'ordre du centimètre.

Par la suite d'autres tests seront effectués sur un système à plus grande échelle.



**8. ANNEXE 1**

Calcul du point milieu entre 2 droites, connaissant pour chaque droite deux points de la droite

*Directement inspiré de <http://www.codes-sources.com/code.aspx?ID=35051>*

```

/*' Calcul de la plus courte distance entre deux droites gauches
'
' Soient A et B deux points de la droite delta 1
' Soient C et D deux points de la droite delta 2
' Soit dt la distance à calculer
' Soit delta' 1 la droite parallèle à delta 1 et déplacée suivant dt de
' telle sorte qu'elle touche la droite delta 2
' On note que dt est la perpendiculaire commune à delta 1 et à delta 2
' dt a donc la direction du produit vectoriel du vecteur AB par le
' vecteur CD
' On note que dt est égal à la projection du vecteur BC sur la direction dt
' Soit dt=produit scalaire du vecteur BC par le vecteur unitaire n
' Ce vecteur unitaire n a la direction et le sens du produit vectoriel
' du vecteur AB par le vecteur CD
' On a vecteur BC=(cx-bx,cy-by,cz-bz).On pose a=cx-bx b=cy-by c=cz-bz
' donc vecteur BC=(a,b,c)
' On a vecteur n=(AB vectoriel CD)/(module de (AB vectoriel CD))
' Avec vecteur AB=(bx-ax,by-ay,bz-az).On pose d=bx-ax e=by-ay f=bz-az
' donc vecteur AB=(d,e,f)
' Avec vecteur CD=(dx-cx,dy-cy,dz-cz).On pose g=dx-cx h=dy-cy i=dz-cz
' donc vecteur CD=(g,h,i)
'
' Calcul de AB vectoriel CD et de son module
'
' AB vectoriel CD = (d,e,f) vectoriel (g,h,i) = (ei-fh,fg-di,dh-eg)
' =(j,k,l)
' donc AB vectoriel CD = (j,k,l)
' Module de AB vectoriel CD = racine carrée de j*j +k*k +l*l =m
' donc module=m
'
' Calcul du vecteur n
'
' vecteur n = (j,k,l)/m =(j/m,k/m,l/m) = (n,o,p)
' donc vecteur n = (n,o,p)
'
' Calcul de dt
'
' dt=BC scalaire n = (a,b,c) scalaire (n,o,p) =an+bo+cp = q
' donc dt=q
*/
a = droite->getP1().x - p2.x;
b = droite->getP1().y - p2.y;
c = droite->getP1().z - p2.z;
d = p2.x - p1.x;
e = p2.y - p1.y;
f = p2.z - p1.z;
g = droite->getP2().x - droite->getP1().x;
h = droite->getP2().y - droite->getP1().y;
i = droite->getP2().z - droite->getP1().z;
j = e * i - f * h;
k = f * g - d * i;
l = d * h - e * g;
m = sqrt(j * j + k * k + l * l);
n = j / m;
o = k / m;
p = l / m;

```

```
q = a * n + b * o + c * p;
dt = q;
```

```
/**
' Soit B' le point issu de B suivant le vecteur n et appartenant à delta'1
' On a vecteur BB'= vecteur n
' On a (B'x-Bx,B'y-By,B'z-Bz)=q(n,o,p)
' Le point B' a pour coordonnées:
' (q*n+Bx,q*o+By,q*p+Bz)=(r,s,t)
*/
r = q * n + p2.x;
s = q * o + p2.y;
t = q * p + p2.z;
/**
' Equation de la droite delta2
,
' soit P un point courant de delta2 : P(x,y,z)
' On peut écrire vecteur CP=K1*vecteur CD
' Soit:x-cx=K1(dx-cx)
'      y-cy=K1(dy-cy)
'      z-cz=K1(dz-cz)
' Soit l'equation de la droite delta2
' (x-cx)/(dx-cx)=(y-cy)/dy-cy)=(z-cz)/(dz-cz)
' ou : (x-cx)/g=(y-cy)/h=(z-cz)/i
,
' Equation de la droite delta'1
,
' Soit P' un point courant de delta'1
' On peut écrire vecteur B'P'=K2*vecteur AB
' Soit:x-r=K2(bx-ax)
'      y-s=K2(by-ay)
'      z-t=K2(bz-az)
' Soit l'équation de la droite delta'1
' (x-r)/(bx-ax)=(y-s)/(by-ay)=(z-t)/(bz-az)
' ou : (x-r)/d=(y-s)/e=(z-t)/f
,
' Intersection des deux droites delta2 et delta'1
,
' Des équations des deux droites on tire :
' z=i(x-cx)/g+cz
' z=f(x-r)/d+t
' Soit (i*x-i*cx+cz*g)/g=(f*x-f*r+t*d)/d
' on pose cz*g-i*cx=u
' et      t*d-f*r=v
*/
u = droite->getP1().z * g - i * droite->getP1().x;
v = t * d - f * r;
/**
' On a : (i*x+u)/g=(f*x+v)/d
' Soit d*i*x+d*u=g*f*x+g*v
'      x(d*i-g*f)=g*v-d*u
*/
x = (g * v - d * u) / (d * i - g * f);
X2 = x;
/**
' dans l'équation de la droite delta2
*/
z = i * (X2 - droite->getP1().x) / g + droite->getP1().z;
Z2 = z;
/**
' de même
*/
y = h * (X2 - droite->getP1().x) / g + droite->getP1().y;
Y2 = y;
```

```
/**
' Calcul de l'autre point de delta 1 ( point H1(x1,y1,z1))
'
' On sait que vecteur H1H2 = vecteur BB'
' soit ; x2-x1=r-bx
'         y2-y1=s-by
'         z2-z1=t-bz
' Donc
*/
X1 = X2 + p2.x - r;
Y1 = Y2 + p2.y - s;
Z1 = Z2 + p2.z - t;
/**
' calcul du point milieu
*/
xm = (X1 + X2) / 2;
ym = (Y1 + Y2) / 2;
zm = (Z1 + Z2) / 2;
```