

ESTIMATION TEMPS-REEL DE LA POSE DE CAMERA BASEE SUR UN MARQUEUR CODE 2D

Mokhtaria BARNOUSSI, Abdelhamid LOUKIL (*), Lahouari KADDOUR EL BOUDADI

Laboratoire de Recherche en Systèmes Intelligents (LARES I)
Département d'Electronique, Faculté de Génie Electrique, USTO-MB
BP.1505, Oran El M'naouer, Oran, ALGERIE
(*) abdelhamid.loukil@univ-usto.dz

Abstract — *Accurate and real-time camera pose estimation is an important issue for applications such as Augmented Reality (AR), SLAMs (Simultaneous Localization and Mapping) in mobile robotics or even Intelligent Video Surveillance (IVS). The presented work concerns the implementation and experimentation of an automatic and real-time pose estimation of camera that will be part of an augmented reality system. The camera pose is estimated thanks to the permanent presence, in the filmed scene, of a marker (coded target) whose position and 3D geometry are perfectly known. In order to reduce the computation time and avoid marker tracking failure, we proposed our own marker. This allowed us to make an accurate alignment and real-time 3D synthetic objects in the real scene.*

Keywords — *Augmented reality / camera calibration / pose estimation / tracking / coded marker.*

Résumé — *L'estimation précise et temps réel de la pose de la caméra est un enjeu important pour des applications telles que la réalité augmentée (RA), les SLAMs (Simultaneous Localization And Mapping) en robotique mobile ou même la vidéo surveillance intelligente (VSI). Le travail présenté dans cet article concerne la réalisation d'une procédure automatique et temps réel d'estimation de pose d'un capteur d'images qui fera partie d'un système de réalité augmentée. Le calcul de la pose est estimé grâce à la présence permanente, dans la scène filmée, d'un marqueur (cible codée) dont la position et la géométrie 3D sont parfaitement connues a priori. Pour des raisons d'allègement du temps de calcul et d'évitement des situations d'échecs du suivi, nous avons proposé notre propre marqueur codé. Cela nous a permis de faire un alignement précis et temps réel des objets synthétiques 3D dans la scène réelle.*

Mots-Clés — *Réalité augmentée / calibration de caméra / estimation de pose / suivi temps réel / marqueur codé.*

Nomenclature :

- $[A]$ *Matrice (3 × 4) exprimant les paramètres internes de la caméra.*
- A, B, C et D *Les 4 sommets du marqueur proposé dans le repère scène 3D.*
- I_A, I_B, I_C , et I_D *Les images (2D) correspondantes aux projetés des 4 sommets 3D du marqueur.*
- $M = [X, Y, Z]^T$ *M - point objet ayant pour coordonnées X, Y, Z dans le repère de la scène 3D (en mètres).*
- $\tilde{M} = [X, Y, Z, 1]^T$ *Représentation en coordonnées homogènes du point M.*
- $m = [u, v]^T$ *m - point image, projeté de M sur l'image 2D et ayant pour coordonnées u et v (en pixels).*
- $\tilde{m} = [u, v, 1]^T$ *Représentation en coordonnées homogènes du point m.*
- $[R, t]$ *Matrice (4 × 4) exprimant le modèle externe de la caméra (pose de la caméra).*
- $[R]$ *Matrice (3 × 3) exprimant la rotation spatiale entre le repère scène et le repère caméra.*
- R_{ij} *Eléments des vecteur unitaires de la matrice rotation (i=1..3, j=1..3).*
- $\vec{t} (t_x, t_y, t_z)^T$ *Vecteur (3 × 1) exprimant la translation entre le repère scène et le repère caméra.*
- u_0, v_0 *Coordonnées du centre optique de la caméra dans le plan image (en pixels).*
- α_u, α_v *Facteurs d'échelle respectivement 'horizontal' et 'vertical' de la caméra (en pixels).*

1. INTRODUCTION

La réalité augmentée (R.A.) est une technique en pleine expansion ces dernières années. Avec la banalisation des moyens de capture d'images ainsi que la multitude des systèmes d'affichages portables, cette discipline a vu ses applications se multiplier de manière vertigineuse et touchant tous les domaines tels que: les jeux vidéos, le cinéma, l'architecture, le tourisme, l'industrie, le domaine militaire, la médecine, la sécurité routière, la robotique mobile et la formation, pour ne citer que ceux-là.

Le principe de la R.A. consiste à ajouter des entités virtuelles à l'image réelle de manière réaliste et en temps réel [1, 2]. Elle comporte deux étapes qui sont exécutées de manière simultanée et en boucle fermée:

- Une étape qui consiste à faire l'acquisition vidéo de la scène 3D (monde réel) à l'aide d'un capteur d'images ainsi que l'estimation temps réel de sa pose.
- Une étape permettant l'incrustation des objets synthétiques (monde virtuel) dans la scène filmée.

Le travail présenté dans cet article concerne l'étape la plus délicate et la plus gourmande en temps de calculs pour un système de réalité augmentée (fig.1), à savoir l'estimation temps réel de la pose de caméra. Son principe consiste à suivre, en permanence dans la scène, un marqueur artificiel (cible codée) dont la position et la géométrie 3D sont parfaitement connues a priori. Pour des raisons d'allègement du temps de calcul et d'évitement des situations d'échecs de suivi (*tracking*), nous avons proposé notre propre marqueur codé.

Le présent article est organisé comme suit: la section 2 présente la procédure de calibration interne de notre caméra. La section 3 est dédiée à l'estimation de pose. Elle comporte 3 parties: une première partie est consacrée à la présentation du marqueur réalisé, une seconde à la phase de détection de ce marqueur dans une séquence vidéo et son identification en temps réel, puis la partie 3, est quant à elle, dédiée à l'estimation de la matrice de passage « repère caméra/repère scène ». Dans la section 4, nous avons présenté la procédure d'incrustation des objets virtuels 3D dans la scène réelle. La section 5 développe l'implémentation informatique, l'expérimentation du système ainsi que sa validation. Nous avons terminé notre article par une conclusion.

2. LA CALIBRATION DE CAMERA

Un capteur d'images sert à faire une projection de la scène réelle 3D sur une image 2D. Cette transformation met en jeu plusieurs phénomènes: optiques (projection centrale à l'aide d'un ensemble de lentilles formant l'objectif), mécaniques (effet de zoom et mise au point) et électroniques (échantillonnage, quantification, et mémorisation de l'image). Le modèle adopté pour modéliser notre camera est celui du sténopé non linéaire (*non-linear pin-hole model*) [3]. Nous pouvons exprimer la position d'un point de l'espace 3D, $M = [X, Y, Z]^T$ en fonction de la position de son image projetée sur le plan image $m = [u, v]^T$ (fig. 2) à l'aide de la relation suivante:

$$s \cdot \tilde{m} = [A] \cdot [R, t] \cdot \tilde{M} \quad (1)$$

Où:

- s : est un facteur d'échelle arbitraire;
- $\tilde{M} = [X, Y, Z, 1]^T$ sont les coordonnées homogènes 3D du point objet M (en mètres);
- $\tilde{m} = [u, v, 1]^T$ sont les coordonnées homogènes 2D du point image m dans le plan de l'image (en pixels);
- $[R, t]$ est la matrice homogène (4×4), appelée « pose de la caméra » et exprimant les paramètres externes de la caméra qui donne la position et l'orientation 3D de la caméra par rapport au repère lié à la scène ($[R]$ matrice rotation et \vec{t} le vecteur translation). Elle est définie par :

$$[R, t] = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

- R_{ij} : sont les éléments des vecteur unitaires de la matrice rotation ;
- $\vec{t} (t_x, t_y, t_z)$: est le vecteur translation représentant la position du centre C de la caméra dans le repère de la scène ;
- $[A]$: est la matrice des paramètres internes de la caméra et est définie par :

$$[A] = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (3)$$

- α_u, α_v : sont les facteurs d'échelle horizontal et vertical respectivement (en pixels) ;
- u_0, v_0 : sont les coordonnées du centre optique dans le plan d'affichage (en pixels).

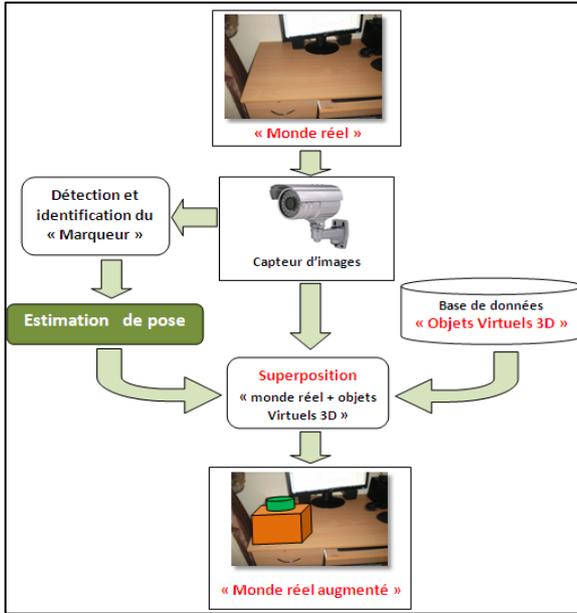


Fig.1 – Synoptique d'un système de réalité augmentée

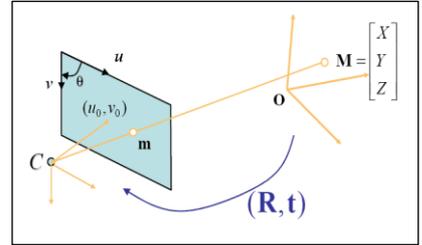


Fig. 2 - Modèle du sténopé non linéaire

Comme cité ci-dessus, la matrice $[R, t]$ représente la pose de la caméra. Cela signifie qu'elle change de valeurs à chaque fois que la caméra bouge. La matrice $[A]$ est, quant à elle, invariante car elle modélise les paramètres physiques de la caméra. L'estimation de cette dernière a été réalisée, hors ligne, à l'aide de l'outil de calibration de caméra du logiciel Matlab « *GML toolbox* » [4]. Cet outil se base sur le modèle du sténopé non linéaire et la méthode de calibration de Zhang [3]. Comme entrées du logiciel, nous avons fourni 6 images de la même mire en damier de (6×5) carrés pour faire la calibration de notre caméra et nous avons obtenu les résultats suivants :

$$[A] = \begin{pmatrix} 416.675 & 0 & 118.605 \\ 0 & 419.614 & 135.145 \\ 0 & 0 & 1 \end{pmatrix}$$

Ce qui signifie que notre caméra possède les caractéristiques physiques suivantes:

- $\alpha_u = 416.675$ pixels;
- $\alpha_v = 419.614$ pixels;
- $u_0 = 118.605$ pixels
- $v_0 = 135.145$ pixels.

Ces paramètres internes de la caméra (matrice $[A]$) seront stockés dans notre base de données et seront utilisés en phase de construction puis affichage de l'image de l'objet synthétique 3D. Dans la section suivante, nous allons présenter la procédure d'estimation de pose de la caméra utilisée.

3. ESTIMATION DE POSE

L'estimation de pose de la caméra, appelée aussi «calibrage extrinsèque de la caméra» est un problème central en vision par ordinateur. Elle permet de calculer la position et l'orientation de la caméra relativement à la scène. Il existe deux méthodes pour l'estimation de pose: une méthode analytique [5, 6, 7] et l'autre itérative [8]. L'avantage des méthodes analytiques est qu'elles utilisent un nombre fixe d'équations et admettent un ensemble fini de solutions mais le résultat obtenu dépend de la qualité des images acquises. Les méthodes itératives permettent d'obtenir une excellente précision pour la localisation de la caméra mais elles présentent l'inconvénient de nécessiter une estimation initiale de la localisation pour converger, de plus, chaque itération exécutée entraîne un surcoût en temps de calcul. J.Y. Didier et F. E. Ababsa [9, 10] ont utilisé une méthode hybride combinant les deux méthodes. En comparant les méthodes itératives avec celles analytiques, nous remarquons que ces dernières ont plus d'avantages par rapport aux premières, surtout en termes de temps de traitement, et elles donnent des résultats précis à condition que les images acquises soient de bonne qualité.

La méthode utilisée pour notre système est inspirée de celle de J.Y. Didier [9]. Elle nécessite la connaissance des paramètres intrinsèques (déjà estimés en section 2) et une mise en correspondance entre les coordonnées 3D de seulement quatre points de référence dont les coordonnées sont parfaitement connus dans le repère de la scène ainsi que de leurs homologues 2D dans le plan d'affichage (repère image).

- ☞ Comment allons-nous disposer de 4 points 3D de référence dans la scène et de leurs projetés dans l'image 2D et ceci de manière permanente?
- ✓ La réponse à cette question a fait l'objet d'une multitude d'études ces vingt dernières années [11, 12, 13, 14]. Elles ont débouché sur deux méthodes :
- Une première méthode consiste à rechercher continuellement dans la scène capturée des objets naturels (amers naturels) connus dont la position et les dimensions sont connues. L'estimation de pose étant réalisée par la mise en correspondance des points caractéristiques de ces objets avec leurs images. Cette méthode porte le nom *d'estimation de pose sans marqueurs (markerless)*

pose estimation). L'inconvénient de celle-ci est le coût en temps de calculs nécessaire au système pour trouver et reconnaître ces amers par rapports aux autres objets de la scène.

- Une deuxième méthode consiste à poser dans la scène, et de manière apparente, un marqueur assez remarquable par rapport aux autres objets de la scène (forme, couleur, texture,...). L'estimation de pose est réalisée par la mise en correspondance des points caractéristiques de ce marqueur avec son image. Cette deuxième méthode porte le nom d'*estimation de pose avec marqueurs* (*marker based pose estimation*). L'avantage de celle-ci est le gain de temps de recherche et de reconnaissance du marqueur qui est facilement discernable par rapport aux objets constituant la scène.

Dans un souci de réaliser un système temps réel, nous nous sommes basé sur la deuxième méthode d'estimation de pose. Dans ce qui suit, nous allons développer dans l'ordre :

- La réalisation du marqueur et son codage 2D,
- La détection et l'identification du marqueur dans l'image.

3.1 Réalisation d'un marqueur et son codage

Le marqueur codé doit être un objet facilement identifiable dans une scène réelle. Il ne doit pas avoir ni symétrie horizontale ni verticale dans le souci de détecter les orientations quelle qu'elle soit. Il existe une multitude de marqueurs disponibles dans des bibliothèques publiques, telles que [15, 16, 17] (Fig. 3) :

- Bibliothèque ARTool-Kit ;
- Bibliothèque Cyber code ;
- Bibliothèque InterSense.



Marqueurs d'ARTool-Kit



Marqueurs d'InterSense

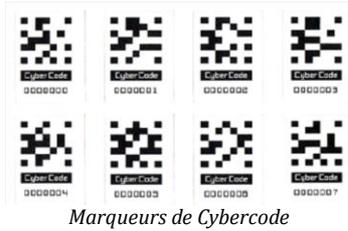


Fig. 3 - Quelques marqueurs disponibles dans des bibliothèques ouvertes (Open Libraries)

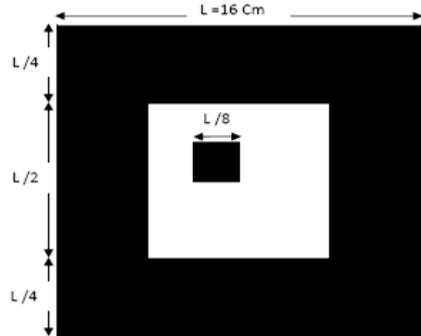


Fig. 4 - Le marqueur LARESI-Marker – Version 0.

Les marqueurs présentés ci-dessus ont l'inconvénient de nécessiter des routines de détection et de reconnaissance assez lourdes en termes de calculs. Nous avons préféré créer notre propre marqueur, que nous avons dénommé « LARESI-Marker », et qui respecte tous les critères suscités.

Comme présenté dans la figure 4, notre marqueur « LARESI-Marker – version 0 » a une forme carrée de 16 cm de côté, de couleur binaire (noir et blanc). Pour éviter la symétrie dans les deux sens (horizontal et vertical), un carré noir de 2 cm de côté est inséré légèrement décalé en haut, à gauche du centre du marqueur. La détection de ce marqueur dans une scène vidéo a été réalisée en testant, dans l'ordre, plusieurs techniques de traitement d'images. Lorsque la scène est non encombrée (Fig. 5), une première procédure de détection du marqueur basée sur une simple binarisation avec un seuil convenablement choisi. La recherche des sommets du marqueur étant faite par un simple balayage de l'image (de haut vers le bas, et de gauche vers la droite). Cette manière de procéder a montré que pour certaines orientations du marqueur, la détection des sommets était erronée (Fig.5.c et Fig.5.d).

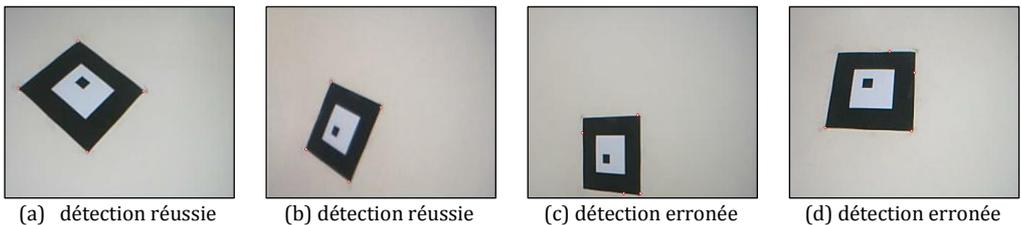


Fig. 5 – Résultats de la procédure de détection du marqueur « LARESI-Marker – Version 0 »

Lorsque la scène est encombrée, une autre technique consiste à appliquer des traitements plus lourds à l'image telles que la recherche de formes dans une image (*Template Matching*) ou bien la détection des coins par la méthode de « Harris ». Ces méthodes ont l'inconvénient d'être très coûteuse en temps de traitement qui va au détriment du critère temps réel de notre système. Dans ce cas aussi, nous avons eu quelques fausses détections des sommets de notre marqueur. Pour pallier à ce problème, nous avons fait une modification de la forme de notre marqueur comme présenté en figure 6.

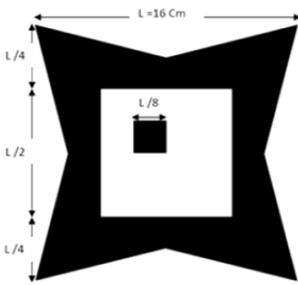


Fig. 6 - Le marqueur amélioré « LARESI-Marker – Version 1 »

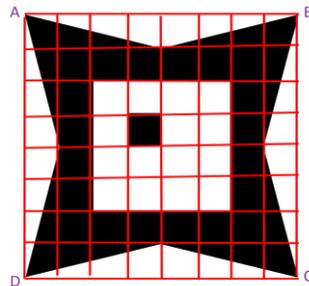


Fig. 7 – Echantillonnage graphique de notre marqueur « LARESI-Marker – Version 1 »

L'utilisation de ce nouveau marqueur nous a permis d'éviter les mauvaises détections des sommets et les résultats obtenus étaient correctes dans toutes les orientations. Une fois les sommets du marqueur détectés, reste maintenant de savoir si c'est notre marqueur qui a été détecté ou bien une forme qui lui ressemble dans la scène filmée. C'est à ce moment que l'on doit mettre en évidence le « codage du marqueur ». Comme illustré dans la figure 7, le marqueur utilisé doit être le plus discernable possible dans une scène encombrée. La manière la plus logique d'aboutir à cette fin est de lui faire correspondre un ou plusieurs codes binaires en fonction de son orientation dans la scène. L'échantillonnage spatial (8×8) de notre marqueur nous fournira un nombre binaire de 64 bits (Fig. 8). Chaque bit est obtenu selon l'état du pixel central de la fenêtre élémentaire (noir = 0 ou blanc =1).



Fig. 8 - Les 4 codes possibles du marqueur selon son orientation (0°, 90°, 180°, 270°)

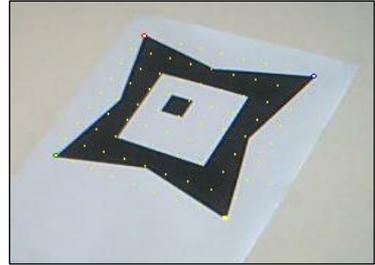


Fig. 9 – Image du Marqueur « LARESI-Marker - Version 1 » filmée en perspective

3.2 Détection et identification du marqueur

Dans une scène vidéo, une fois la détection des sommets réalisée, nous entamons la phase d'identification. Cette dernière consiste à faire un échantillonnage spatial de l'objet détecté (marqueur ou non marqueur) ensuite rechercher le code correspondant puis le comparer avec les 4 codes possibles (Fig. 8) de notre marqueur.

Les objets filmés sont généralement visibles en perspective et non de manière orthogonale à la caméra (Fig. 9), ceci implique que l'échantillonnage de notre forme visible doit prendre en compte l'angle de vue de la caméra ainsi que la profondeur appelée aussi facteur d'échelle de celle-ci (l'effet de zoom).

Deux phases sont nécessaires pour l'échantillonnage ainsi que le codage du marqueur filmé :

- Une première phase consiste à calculer les 4 distances entre le centre optique de la caméra et chacun des 4 sommets du marqueur (A, B, C et D sur la fig.7).
- Une deuxième phase consiste à calculer la position des images (u_i, v_i) des 64 points d'échantillonnage en respectant l'effet de perspective.

3.2.1 **Première phase:** Calcul des distances « centre optique de la caméra / 4 sommets du marqueur »

Etant données les valeurs des coordonnées images de chaque sommet du marqueur (en pixels): $I_A(u_A, v_A)$, $I_B(u_B, v_B)$, $I_C(u_C, v_C)$ et $I_D(u_D, v_D)$ ainsi que les coordonnées 3D réelles de chaque sommet dans le repère lié à la caméra (en mètres): $A(X_A, Y_A, Z_A)$, $B(X_B, Y_B, Z_B)$, $C(X_C, Y_C, Z_C)$ et $D(X_D, Y_D, Z_D)$. La relation liant l'objet à son image est donnée par l'équation (1) que nous pouvons réécrire comme suit pour le sommet 'A' (de même pour les 3 autres sommets 'B', 'C' et 'D'):

$$\begin{cases} u_A = \alpha_u \cdot \frac{X_A}{Z_A} + u_0 \\ v_A = \alpha_v \cdot \frac{Y_A}{Z_A} + v_0 \end{cases} \quad (4)$$

Les inconnues dans ce dernier système sont les profondeurs, c.à.d. pour les 4 sommets, les ordonnées Z_A, Z_B, Z_C et Z_D . Aussi, du fait que notre marqueur est plan et de forme carrée, l'équation vectorielle suivante est toujours vérifiée dans l'espace 3D (Fig. 7) :

$$\overrightarrow{AB} = \overrightarrow{DC} \quad (5)$$

En réécrivant cette dernière équation en fonction des coordonnées 3D tirées de l'équation (4), et en forçant un paramètre $Z_A = 1$ (normalisation des profondeurs par rapport à la profondeur du sommet A), nous obtenons les 3 profondeurs restantes comme suit:

$$\begin{cases} Z_B = \frac{1}{\det M} \cdot [u_A(v_D - v_C) + v_A(u_C - u_D) - u_C v_D + u_D v_C] \\ Z_C = \frac{1}{\det M} \cdot [u_A(v_D - v_B) + v_A(u_B - u_D) - u_B v_D + u_D v_B] \\ Z_D = \frac{1}{\det M} \cdot [u_A(v_C - v_B) + v_A(u_B - u_C) - u_B v_C + u_C v_B] \end{cases} \quad (6)$$

Avec : $\det M = u_B(v_D - v_C) + u_C(v_B - v_D) + u_D(v_C - v_B)$.

3.2.2 Deuxième phase: Calcul de la position des images (u_i, v_i) des 64 points d'échantillonnage en respectant l'effet de perspective.

En appliquant la formule barycentrique de l'échantillonnage simple aux coordonnées (X_{ij}, Y_{ij}, Z_{ij}) des 64 points échantillonnés ($i = 1$ à m et $j = 8$ à n), nous aurons:

$$\begin{cases} X_{ij} = \frac{1}{m \cdot n} [X_A(m-i)(n-j) + X_B i(n-j) + X_C i j + X_D(m-i)j] \\ Y_{ij} = \frac{1}{m \cdot n} [Y_A(m-i)(n-j) + Y_B i(n-j) + Y_C i j + Y_D(m-i)j] \\ Z_{ij} = \frac{1}{m \cdot n} [Z_A(m-i)(n-j) + Z_B i(n-j) + Z_C i j + Z_D(m-i)j] \end{cases} \quad (7)$$

Pour notre cas, l'échantillonnage 2D du marqueur est de dimension (8×8), c.à.d: $m = 8$ et $n = 8$.

En fusionnant les équations (4), (6) et (7) et après simplification, nous aboutirons à :

$$\begin{cases} u_{ij} = \frac{1}{m \cdot n \cdot Z_{ij}} [u_A Z_A(m-i)(n-j) + u_B Z_B i(n-j) + u_C Z_C i j + u_D Z_D(m-i)j] \\ v_{ij} = \frac{1}{m \cdot n \cdot Z_{ij}} [v_A Z_A(m-i)(n-j) + v_B Z_B i(n-j) + v_C Z_C i j + v_D Z_D(m-i)j] \end{cases} \quad (8)$$

Pour tester la robustesse de nos deux procédures ci-dessus, nous avons appliqué ces formules pour échantillonner un damier filmé en perspective ainsi que notre marqueur. Les résultats obtenus (Fig. 10 et Fig. 11) confirment et valident les procédures implémentées.

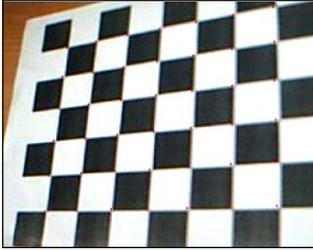


Fig. 10- Résultat de l'échantillonnage perspectif appliqué sur un damier filmé en perspective.



Fig. 11- Résultat de l'échantillonnage perspectif appliqué sur notre marqueur.

Une fois les points échantillonnés localisés dans le marqueur détecté, il suffit de lire la couleur du pixel correspondant pour lui attribuer sa valeur binaire (0 ou 1). Le code détecté du marqueur filmé est alors obtenu en concaténant la valeur binaire de tous les pixels. Si le code obtenu coïncide avec une des valeurs décimales suivantes: « 64511-65471-65503-65023 » (Fig. 8), alors le marqueur est identifié comme étant le « bon », sinon l'objet filmé ne peut être traité comme un marqueur, et ne peut contribuer à l'estimation de pose de notre caméra.

3.3 Estimation de pose de la caméra

Les résultats obtenus dans la phase précédente de la mise en correspondance 3D/2D (équations 6 et 7) seront utilisés pour localiser la caméra par rapport au repère lié à la scène. Sachant que les dimensions du marqueur sont parfaitement connues, et de ce fait les normes des vecteurs liant les extrémités A, B, C et D le sont aussi, nous pouvons exploiter ces données pour calculer les paramètres extrinsèques de notre caméra.

Le repère scène d'origine 'O' est le repère qui est lié au marqueur (Fig. 2). Son origine est au centre de celui-ci, son orientation est de telle manière que le plan (OXY) soit confondu avec le plan du marqueur, l'axe (OZ) est obtenu de telle manière que le repère soit orthogonal et direct, c'est-à-dire sortant du plan du marqueur (Fig. 12).

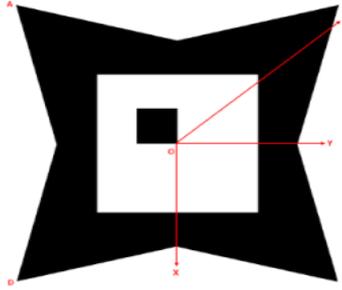


Fig. 12- Position et orientation du repère lié à la scène.

L'estimation de pose de notre caméra est la procédure mathématique qui consiste à estimer les 12 paramètres de la matrice $[R,t]$ de l'équation (2). Pour ce faire, il suffit d'exprimer les normes des vecteurs \overrightarrow{AC} et \overrightarrow{BD} , qui sont, rappelons le, parfaitement connus.

$$\|\overrightarrow{AC}\| = \sqrt{(X_c - X_A)^2 + (Y_c - Y_A)^2 + (Z_c - Z_A)^2} \quad (9)$$

$$\|\overrightarrow{BD}\| = \sqrt{(X_D - X_B)^2 + (Y_D - Y_B)^2 + (Z_D - Z_B)^2} \quad (10)$$

En plus de ces deux dernières équations, nous faisons intervenir les équations de la projection centrale (1) et (4). Les coordonnées image des 4 sommets du marqueur étant connus (équation 8), nous obtenons alors les 12 paramètres de notre matrice $[R,t]$.

4. INCRUSTATION D'OBJETS VIRTUELS 3D DANS LA SCENE REELLE

Dans les sections précédentes, nous avons explicité toutes les procédures de calibration interne de la caméra (estimation de la matrice $[A]$), la détection et l'identification du marqueur dans la scène filmée, puis l'estimation de pose de notre caméra par rapport à notre marqueur (estimation de la matrice $[R,t]$).

Arrivé à ce point, nous pouvons exploiter l'équation (1) du modèle du sténopé non linéaire de manière directe. Cela veut dire, si l'on dispose des coordonnées 3D d'un point objet situé dans la scène, nous pouvons, sans difficulté, calculer la position de son image. C'est cette procédure qui a été utilisée pour incruster un objet virtuel 3D, présent dans une base de données, directement et en temps réel, sur l'image issue de la caméra. Pour que cette procédure ne soit pas gourmande en temps de calcul, et de ce fait garantir le critère temps réel de notre système, nous avons créé dans la base de données de scènes virtuelles, deux objets 3D en modèle « fils de fer » ayant les caractéristiques suivantes :

Objet 1 :

- Forme : Cube;
- Couleur :Rouge ;
- Dimensions : 16 cm × 16 cm × 10 cm.

Objet 2 :

- Trois axes formant un repère direct orthogonal ;
- Axe OX : couleur rouge ;
- Axe OY : couleur bleue ;
- Axe OZ : couleur jaune.

Les résultats de l'incrustation temps réel des deux objets sont présentés dans la figure 13.

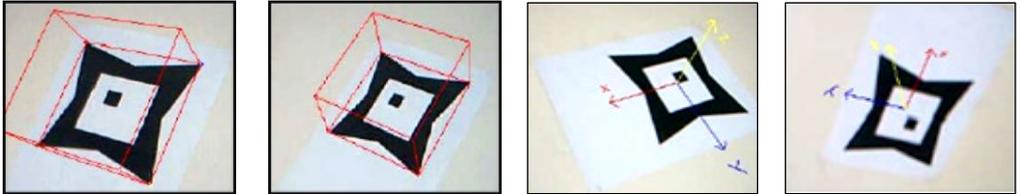


Fig. 13 - Incrustation des deux objets virtuels 3D

5. IMPLEMENTATION ET EVALUATION EXPERIMENTALE

L'implémentation logicielle de toutes les procédures développées précédemment a été réalisée en Builder C++ Version 6. Les figures 14 et 15 sont des captures d'écran de l'interface graphique réalisée.

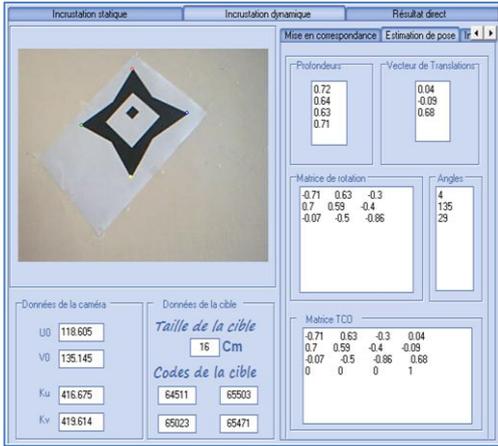


Fig. 14 - Exécution de la procédure d'estimation de pose

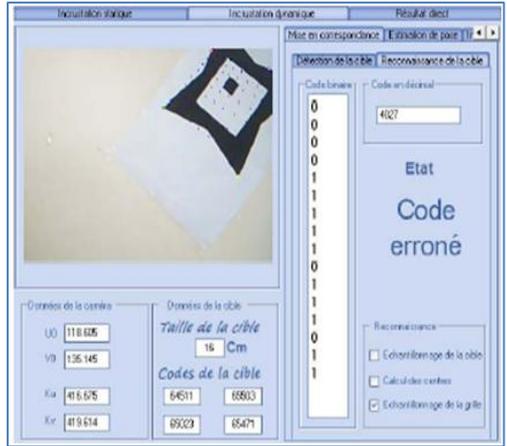


Fig. 15 - Exécution de la procédure d'identification d'un marqueur tronqué

Lors de l'exploitation pratique de notre système, nous avons relevé quelques limitations d'utilisation qui sont énumérées ci-après :

- La meilleure cadence d'images obtenue par notre système de réalité augmentée est de 17 images par seconde, qui est en elle-même suffisante pour un bon suivi et alignement des images synthétiques superposées aux images réelles.
- La phase de détection du marqueur ne donne des résultats probants que si la distance « caméra/ marqueur » est dans l'intervalle [0.36 m, 4.82 m]. Ceci est dû à la valeur de la focale de l'objectif installé sur notre caméra. Plus proche de 0.36m, le marqueur apparaît tronqué (incomplet) sur l'image, plus loin que 4.82 m, il n'est plus détecté car trop petit.

6. CONCLUSION

Le travail présenté dans cet article concerne l'implémentation matérielle et logicielle d'une technique d'estimation de pose temps-réel d'une caméra qui sera intégrée dans un système de réalité augmentée.

Nous avons commencé par la calibration des paramètres internes de notre caméra à l'aide de l'outil « *GML Toolbox* » du logiciel Matlab. L'estimation de pose a nécessité une mise en correspondance 3D-2D en temps réel. La méthode proposée pour la mise en correspondance 3D-2D se base sur la détection et l'identification de notre marqueur, baptisé « LARESI-Marker », qui devra être présent en permanence dans la scène réelle. Dans le but de valider la robustesse

et la précision de notre technique, la '*pose de la caméra*' estimée a été utilisée pour incruster des objets virtuels 3D dans les images de la scène réelle, en temps réel.

7. REMERCIEMENTS

Avant tout, nous tenons à remercier le comité de lecture de la revue « COST » pour ses remarques et ses directives constructives ayant permis l'amélioration de notre article.

Le travail présenté dans cet article a été réalisé au sein du laboratoire de recherche en systèmes intelligents (LARESI) sis à l'USTO-MB d'Oran (Algérie) et rentre dans le cadre d'un projet de Magister. Que toutes les personnes qui ont contribuées de près ou de loin à l'élaboration de ce travail en soient remerciées.

8. REFERENCES

- [1] R. Azuma, Y. Baillet, Y. Behringer, R. Feiner, S. Julier, and B. MacIntyre, *Recent advances in augmented reality*, IEEE Computer Graphics and Applications, Vol. 21, N°6, 2001, pp.34-47.
- [2] D. W. F. Van Krevelen, R. Poelman, *A Survey of augmented reality technologies, Applications and Limitations*, International Journal of Virtual Reality, Vol. 9, N°2, 2010, pp.1-20.
- [3] Z. Zhang, *A flexible new technique for camera calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, N°11, 2000, pp.1330-1334.
- [4] GML Camera Calibration Toolbox official site :
<http://gml-camera-calibration-toolbox.software.informer.com>
- [5] M.A. Ameller, L. Quan, B. Triggs, *Camera Pose Revisited : new linear algorithms*, Proceedings of the RFIA'02 (Reconnaissance des Formes et Intelligence Artificielle), France, 2002, pp.39-47.
- [6] R. Hartley, A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.
- [7] T. Nöll, A. Pagani, D. Stricker, *Real-time camera pose estimation using correspondences with high outlier ratios*, In VISAPP'2010, International Conference on Computer Vision Theory and Applications, 2010, pp.381-386.
- [8] C. P. Lu, G. D. Hager, E. Mjolsness, *Fast and globally convergent pose estimation from video images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, n°6, 2000, pp.610-622.

- [9] F.E. Ababsa, M. Mallem, *Robust Camera Pose Estimation Using 2D Fiducials Tracking for Real-Time Augmented Reality Systems*, Proc. of the 2004 ACM-SIGGRAPH, International conference on Virtual Reality continuum and its applications in industry, USA, 2004, pp.431-435.
- [10] J. Y. Didier, *Contributions à la dextérité d'un système de réalité augmentée mobile appliqué à la maintenance industrielle*, Thèse de doctorat de l'Université d'Évry Val d'Essonne, Évry, France, Décembre 2005.
- [11] M. A. Abidi, T. Chandra, *A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and Evaluation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, N°5, 1995, pp.534-538.
- [12] H. Kato, M. Billinghurst, *Marker tracking and HMD calibration for a video-based augmented reality conferencing system*, Proceedings of the 2nd IEEE and ACM International workshop on Augmented Reality (IWAR'99), 1999, pp.85-94.
- [13] J. Chandaria, G.A. Thomas, D. Stricker, *The MATRIS project: real-time markerless camera tracking for augmented reality and broadcast application*, Journal of Real-Time Image Processing, Vol. 2, N°2-3, 2007, pp.69-79.
- [14] T. Nöll, A. Pagani, D. Stricker, *Markerless Camera Pose Estimation – An Overview*, Proceedings of VLUDS, Vol. 19, Germany, 2010, pp.45-54.
- [15] <http://www.hitl.washington.edu/artoolkit/>
- [16] <http://www.intersense.com>
- [17] J. Rekimoto, Y. Ayatsuka, *CyberCode: designing augmented reality environments with visual tags*, Proceedings of DARE 2000 on Designing augmented reality environments, ACM, USA, 2000, pp.1-10.
- [18]