Caméras virtuelles pour la calibration d'un système de réalité augmentée composé d'un écran transparent et deux caméras à champs disjoints

Jim Braux-Zin, Adrien Bartoli, Romain Dupont, Régis Vinciguerra

Caméras virtuelles pour la calibration d'un système de réalité augmentée composé d'un écran transparent et deux caméras à champs disjoints

Jim Braux-Zin[1] Adrien Bartoli[2] Romain Dupont[1] Régis Vinciguerra[1]

[1] CEA, LIST, 91191 Gif-sur-Yvette, France
[2] ISIT, Université d'Auvergne, 63000 Clermont-Ferrand, France

jim.braux-zin@cea.fr

Résumé

Une nouvelle méthode de calibration extrinsèque pour systèmes de réalité augmentée sur affichages transparents est proposée. Elle est principalement destinée à un système composé d'un écran semi-transparent, d'une caméra de suivi de l'utilisateur et d'une autre caméra, à champ disjoints, pour localiser le système au sein de l'environnement mais facilement généralisable. L'algorithme se base sur des indications fournies par l'utilisateur sur la projection apparente sur l'écran de points de référence d'un objet connu. Une estimation convexe est calculée grâce à la calibration de caméras virtuelles et sert d'initialisation à un ajustement de faisceaux global. Des expériences sur données synthétiques et réelles montrent le bien-fondé de cette approche. Ceci est une traduction de la publication originale [3].

Mots Clef

calibration, réalité augmentée, écran transparent, caméras à champs disjoints

Abstract

We present a novel extrinsic calibration method for optical see-through systems. It is primarily aimed at tablet-like systems with a semi transparent screen, a camera tracking the user position and another camera analyzing the scene but easily generalizable to any optical see-through setup. Relative poses of the cameras and the screen are all needed for proper alignment. The proposed algorithm is based on the user indicating the projections onto the screen of several reference points chosen on a known object. A convex estimation is computed through the resectioning of virtual cameras and used to initialize a global bundle adjustment. Both synthetic and real experiments show the viability of our approach. This is a translation of the original publication [3].

Keywords

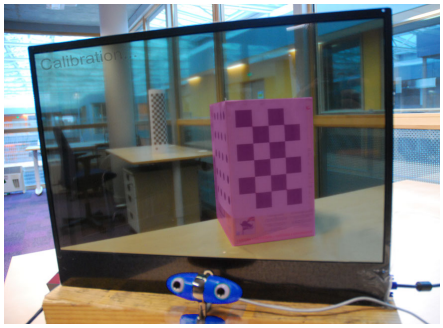
calibration, augmented reality, optical see-through, non-overlapping cameras

Introduction

Les systèmes de réalité augmentée classiques superposent des éléments virtuels sur le flux vidéo d'une scène réelle. Le faible coût du matériel nécessaire et la simplicité de la technique ont permis un développement rapide. Cependant, les applications critiques telles que l'assistance aux opérations chirurgicales ou l'aide à la conduite ne peuvent tolérer aucune indirection entre la réalité et l'utilisateur. Des systèmes utilisant un affichage semi-transparent sont plus adaptés à ces cas et permettent d'améliorer l'immersion pour les applications plus classiques. Jusqu'à présent ces systèmes sont restés limités à des usages de niche. Par exemple les affichages tête haute présents dans certains avions et composés d'un projecteur, de lentilles et de miroirs sont chers, encombrants et souvent restreints à un affichage monochrome. Un intérêt industriel croissant s'affiche pour les lunettes semi-transparentes mais malgré des progrès indéniables ces systèmes restent trop intrusif et sujets aux rapides mouvements de tête rendant l'analyse de la scène difficile.

Nous nous intéressons ici à une approche «tablette» utilisant un écran LCD (Figure 1) auxquels deux caméras sont fixées et dirigées vers des côtés opposés. Cela permet de s'affranchir de la plupart des inconvénients des systèmes cités précédemment tout en restant mobile et flexible. La caméra frontale suit l'utilisateur pendant que l'autre caméra analyse la scène pour se localiser. Les poses de ces caméras par rapport à l'écran sont nécessaires pour aligner l'affichage avec la réalité, mais les champs de vision sont disjoints et aucune caméra ne voit l'écran. Les méthodes classiques de calibration extrinsèques utilisant un motif connu pour la localisations sont donc inapplicables.

La calibration extrinsèque de deux caméras à champs disjoints a déjà été traitée dans la littérature. Plusieurs méthodes se basent sur la localisation des caméras mobiles [4, 6, 10] ou statiques [2, 12] mais ne permettent aucunement d'estimer la pose de l'écran. À notre connaissance, les seules méthodes permettant d'obtenir toutes les poses requises utilisent un miroir pour calculer la pose d'un objet hors du champ de vision d'une caméra [9, 13, 15, 16]. En appliquant une de ces méthodes pour estimer la pose de



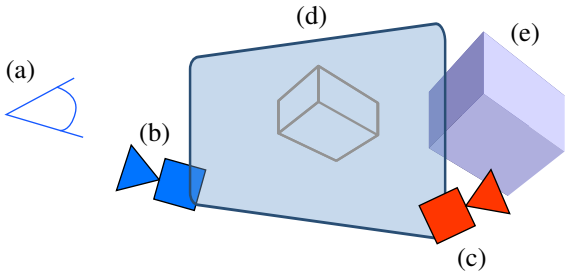


FIGURE 1 – Système composé de : (a) utilisateur, (b) caméra de suivi de visage, (c) caméra d'analyse de la scène, (d) écran semi-transparent, (e) scène

chaque caméra par rapport à l'écran, il est théoriquement possible de calibrer complètement le système.

Le problème est que ces méthodes minimisent l'erreur de reprojection dans le plan image de la caméra des images de la cible (ici l'écran) à travers le miroir. Pour l'utilisateur, l'important est de réduire l'*erreur d'alignement*, c'est à dire la distance à l'écran entre l'augmentation virtuelle et la scène réelle. Il s'agit en effet de l'objectif de la plupart des méthodes de calibration des systèmes de type lunettes [17]. En considérant le système dans son ensemble l'approche est indépendante des capteurs et algorithmes utilisés pour le suivi. On peut par exemple utiliser sans problème une caméra 3d ou un système électromagnétique pour le suivi de l'utilisateur. De plus, à la différence des méthodes précédentes considérant chaque caméra indépendamment, les erreurs des capteurs ou de modélisations peuvent être compensées. Nous proposons une nouvelle méthode de minimisation de l'erreur d'alignement dans le contexte du scénario de calibration suivant.

Un objet connu est placé derrière l'écran, dans le champ de la caméra. La position de plusieurs points clés de l'objet dans le repère de cette caméra est connue (SLAM, relocalisation…). L'utilisateur indique la projection apparente de ces points sur l'écran pendant que la deuxième caméra suit sa position. Ce processus est pour le moment effectué en cliquant sur les projections dans un ordre prédéfini et depuis différents points de vue.

Énoncé du problème et plan Le processus de calibration, formalisé Section 1, doit minimiser la distance entre l'affichage et les clics de l'utilisateur. Cette fonction de coût est non convexe à cause de la présence de deux rotations. Pour garder la méthode générique [1], aucune hypothèse n'est faite sur les poses des caméras et l'initialisation est effectuée grâce à une approximation convexe robuste basée sur l'introduction de caméras virtuelles. Ce processus est expliqué Section 2 ainsi que l'étape d'ajustement de faisceaux qui suit. Enfin la Section 3 est dédiée à l'évaluation de la précision de la méthode sur des données synthétiques et réelles.

1 Formulation du problème

Les notations utilisées sont les suivantes : les points et vecteurs 3d sont représentés par des lettres majuscules $X = (X_x, X_y, X_z)^T$, les points et vecteurs 2d par des lettres minuscules x, les matrices par des lettres majuscules en gras $\mathbf{M}$ et les caméras par des lettres calligraphiées $\mathcal{C}$. La même notation est utilisée pour désigner la caméra et son repère associé. Le repère monde est noté $\mathcal{W}$. Nous utiliserons la norme de Mahalanobis notée $||x||_\Sigma = \sqrt{x^T \Sigma^{-1} x}$.

Trois composants sont fixés rigidement dans la configuration choisie : l'écran semi-transparent, la caméra $\mathcal{C}_u$ tournée vers l'utilisateur et la caméra $\mathcal{C}_s$ tournée vers la scène. L'écran transparent est utilisé comme référence et définit le repère monde $\mathcal{W}$ où les axes sont tels que Figure 2a et l'origine un point arbitraire de l'écran. La pose de $\mathcal{C}_u$ dans $\mathcal{W}$ s'exprime par la transformation $\mathbf{M}_u = \begin{bmatrix}\mathbf{R}_u & T_u\end{bmatrix}$ où $\mathbf{R}_u$ est la matrice de rotation 3×3 et T_u le vecteur translation 3×1. De même, $\mathbf{M}_s$, $\mathbf{R}_s$, et T_s décrivent la pose de $\mathcal{C}_s$.

Les m positions de l'utilisateur dans $\mathcal{C}_u$ sont notées U_i, $i = 1 \dots m$. Les n points clés de l'objet dans $\mathcal{C}_s$ sont notés O_j, $j = 1 \dots n$. Dans le repère monde, elles sont respectivement $(\mathbf{M}_u U_i)_{i=1\dots m}$ et $(\mathbf{M}_s O_j)_{j=1\dots n}$.

Pour une position utilisateur U_i et un point clé O_j, c_{ij} est l'intersection du rayon $(\mathbf{M}_u U_i, \mathbf{M}_s O_j)$ avec le plan $z = 0$ (l'écran) dans $\mathcal{W}$. On définit f_{int} la fonction d'intersection :

$$f_{\text{int}}\left(\begin{bmatrix}x_1\\y_1\\z_1\end{bmatrix}, \begin{bmatrix}x_2\\y_2\\z_2\end{bmatrix}\right) = \begin{pmatrix} x_1 - z_1 \times \frac{x_2 - x_1}{z_2 - z_1} \\ y_1 - z_1 \times \frac{y_2 - y_1}{z_2 - z_1} \end{pmatrix} \quad (1)$$

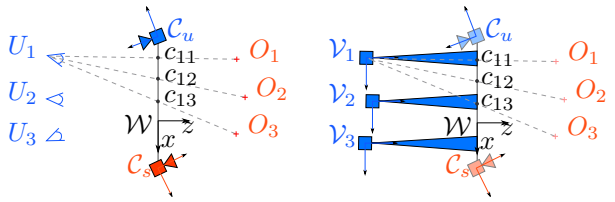
pour $z_1 > 0$ et $z_2 < 0$ donc :

$$c_{ij} = f_{\text{int}}(\mathbf{M}_u U_i, \mathbf{M}_s O_j) \quad (2)$$

Les coordonnées de c_{ij} sont exprimées en unités du monde : la conversion depuis les coordonnées pixelliques est triviale à partir des dimensions de l'écran.

Modèle de bruit L'estimation des positions de l'utilisateur et des points clés de l'objet sont sujettes à du bruit. Les informations fournies par l'utilisateur sont également imprécises. Ces bruits sont modélisés par des variables gaussiennes dont les matrices de covariances sont diagonales (bruit indépendant dans chaque dimension). On note avec

1. Plusieurs généralisations intéressantes sont présentées Section 2.6



(a) (b)

FIGURE 2 – Vue de dessus du système : (a) notations (les éléments de même couleur sont exprimés dans le même repère), (b) définitions des caméras virtuelles utilisateur (voir texte).

un tilde les versions bruitées des variables d'entrée du système :

$$\widetilde{U}_i = U_i + n(0, \mathbf{\Sigma}_U) \tag{3}$$

$$\widetilde{O}_j = O_j + n(0, \mathbf{\Sigma}_O) \tag{4}$$

$$\widetilde{c}_{ij} = c_{ij} + n(0, \mathbf{\Sigma}_c) \tag{5}$$

pour tout i entre 1 et m et j entre 1 et n. Les bruits varient peu pour une configuration donnée et il est possible d'estimer leurs covariances (voir Appendice B pour un exemple). Cette information permet d'améliorer la précision de la calibration. Il est nécessaire de distinguer les paramètres estimés de leur véritable valeur. On note les estimations avec un chapeau : $\widehat{\mathbf{M}}_u$, $\widehat{\mathbf{M}}_s$, $\widehat{U}_{i=1\ldots m}$, $\widehat{O}_{j=1\ldots n}$.

Problème de calibration L'objectif de la méthode proposée est de trouver les meilleures estimations $\widehat{\mathbf{M}}_u$ et $\widehat{\mathbf{M}}_s$ sachant $\widetilde{U}_{i=1\ldots m}$, $\widetilde{O}_{j=1\ldots n}$ et les $\widetilde{c}_{ij}$ correspondants, grâce à l'équation (2). En considérant des bruits gaussiens, la solution optimale minimise l'ajustement de faisceaux avec covariances suivant :

$$\begin{aligned} f(\mathbf{M}_u, \mathbf{M}_s, U_{i=1\ldots m}, O_{j=1\ldots n}) = \\ \sum_{\substack{i=1\ldots m \\ j=1\ldots n}} \sum \|\widetilde{c}_{ij} - f_{\text{int}}(\mathbf{M}_u U_i, \mathbf{M}_s O_j)\|^2_{\mathbf{\Sigma}_c} \\ + \sum_{i=1\ldots m} \left\|\widetilde{U}_i - U_i\right\|^2_{\mathbf{\Sigma}_U} + \sum_{j=1\ldots n} \left\|\widetilde{O}_j - O_j\right\|^2_{\mathbf{\Sigma}_O} \end{aligned} \tag{6}$$

Le premier terme est l'erreur d'alignement 2d, non convexe. La section suivante présente une méthode originale d'initialisation convexe.

2 Calibration et caméras virtuelles

2.1 Définition des caméras virtuelles

L'introduction de caméras virtuelles permet de décomposer le problème en plusieurs calibrations classiques. On définit m caméras virtuelles $\mathcal{V}_i$ centrées sur les positions utilisateurs et partageant toutes le plan de l'écran ($z = 0$) comme plan focal. Leurs points principaux sont confondus avec l'origine de $\mathcal{W}$. Ces caméras ont des propriétés intéressantes : tous leurs axes optiques sont parallèles et pointent vers la direction $+z$ dans $\mathcal{W}$. Les poses des caméras sont donc définies par, pour tout i entre 1 et m :

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{I} & T_i \end{bmatrix} \tag{7}$$

où $\mathbf{I}$ est la matrice identité 3×3 et $T_i = \mathbf{M}_u U_i$ est la position de U_i dans $\mathcal{W}$. Comme les c_{ij} sont exprimés en unités de $\mathcal{W}$, les «pixels» des caméras virtuelles sont parfaitement carrés : les longeurs focales en x et y sont égales et l'obliquité est nulle. De plus, comme les caméras sont définies par leur plan focal et point principal dans $\mathcal{W}$, leurs paramètres intrinsèques sont liés à la position de leur centre optique.

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & u_i \\ 0 & f_i & v_i \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -T_{iz} & 0 & T_{ix} \\ 0 & -T_{iz} & T_{iy} \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Les c_{ij} sont les projections des points objet dans $\mathcal{V}_i$, donc en coordonnées homogènes :

$$c_{ij} \propto \mathbf{K}_i \mathbf{M}_i^{-1} \mathbf{M}_s O_j \tag{9}$$

$$c_{ij} = \lambda_{ij} \mathbf{H}_i O_j \quad \lambda_{ij} \in \mathbb{R} \quad \forall j \in 1 \ldots n \tag{10}$$

où

$$\mathbf{H}_i = \mathbf{K}_i \mathbf{M}'_i \text{ and } \mathbf{M}'_i = \mathbf{M}_i^{-1} \mathbf{M}_s = [\mathbf{R}_s | T_s - T_i] \tag{11}$$

2.2 Calibration des caméras virtuelles

Nous verrons que le calcul des paramètres extrinsèques et intrinsèques d'une caméra virtuelle $\mathcal{V}_i$ permet d'estimer la pose de la caméra réelle $\mathcal{C}_s$ qui observe la scène. Ce processus de calibration est détaillé dans [7]. On utilise l'approche Direct Linear Transform (DLT) présentée dans [1] pour estimer $\mathbf{H}_i$ à partir de l'équation (10). Le facteur scalaire est éliminé grâce à un produit vectoriel.

$$c_{ij} \wedge \mathbf{H}_i O_j = 0 \quad \forall j \in 1 \ldots n \tag{12}$$

Ces équations vectorielles donnent $2 \times n$ équations linéaires indépendantes sur les coefficients de $\mathbf{H}_i$. Avec $n \geq 6$ points clés sur l'objet, il est possible de calculer une solution aux moindres carrés pour les 12 coefficients avec une décomposition en valeurs singulières. La solution obtenue minimise la distance algébrique et non la distance euclidienne mais constitue une bonne initialisation.
L'étape suivante est d'extraire les paramètres intrinsèques $\mathbf{K}_i$ et extrinsèques $\mathbf{M}'_i$ tels que $\mathbf{H}_i = \mathbf{K}_i \mathbf{M}'_i$. On part d'une décomposition directe avec l'équation

$$\overline{\mathbf{H}}_i \overline{\mathbf{H}}_i^T = \mathbf{K}_i \mathbf{R}_s \mathbf{R}_s^T \mathbf{K}_i^T = \mathbf{K}_i \mathbf{K}_i^T \tag{13}$$

où $\overline{\mathbf{H}}_i$ est la sous-matrice 3×3 de $\mathbf{H}_i$. Dans le cas de données bruitées, cette décomposition donne une matrice de paramètres intrinsèques génériques ne respectant pas les propriétés de eq. (8) :

$$\mathbf{K}_i^{\text{init}} = \begin{bmatrix} f_{xi} & s_i \cdot f_{xi} & u_i \\ 0 & f_{yi} & v_i \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Cette décomposition initiale est raffinée itérativement avec une contrainte douce [7] pour forcer la matrice intrinsèque à prendre la forme désirée. L'algorithme de Levenberg-Marquardt est utilisé pour minimiser :

$$f(\mathbf{K}_i, \mathbf{M}'_i, O_{j=1\dots n}) = \sum_{j=1\dots n} \|\widetilde{c}_{ij} - \lambda_{ij}\mathbf{K}_i\mathbf{M}'_i O_j\|^2_{\mathbf{\Sigma}_c} + \sum_{j=1\dots n} \left\|\widetilde{O}_j - O_j\right\|^2_{\mathbf{\Sigma}_O} + w\left(|f_{xi} - f_{yi}| + |s|\right) \tag{15}$$

où w est un poids qui doit augmenter lentement à chaque itération. En pratique, nous avons observé qu'il suffit d'effectuer une optimisation jusqu'à convergence avec $w = \frac{1}{\sigma_{\text{min}}}$, où σ_{min} est le plus petit coefficient diagonal de $\mathbf{\Sigma}_c$ et $\mathbf{\Sigma}_O$. Ce choix permet de s'assurer que la contrainte a un impact indépendamment du niveau de bruit.

2.3 Extraction de la pose de la caméra $\mathcal{C}_s$

Il est maintenant possible d'extraire T_i à partir de $\mathbf{K}_i$ grâce à l'équation (8). On construit ensuite $\mathbf{M}_i$ à partir de T_i en utilisant l'équation (7). Enfin on extrait $\widehat{\mathbf{M}}_s^{(i)}$ de $\mathbf{M}'_i$ avec l'équation (11).

Les différentes estimations $\widehat{\mathbf{M}}_s^{(i)}$ doivent être agrégées en un $\widehat{\mathbf{M}}_s$ optimal pour initialiser l'ajustement de faisceaux. On élimine dans un premier temps les échecs manifestes : caméra virtuelles pour lesquelles le raffinement non linéaire n'a pas convergé où dont les matrices de paramètres intrinsèques sont invalides. Ensuite, nous avons choisi de sélectionner l'estimation dont l'erreur (16) après raffinement est la plus faible. C'est l'heuristique qui donne les meilleurs résultats selon notre expérience.

2.4 Estimation de la caméra $\mathcal{C}_u$

Les caméras virtuelles sont centrées sur les positions de l'utilisateur et la calibration de chaque $\mathcal{V}_i$ donne une estimation T_i de $\mathbf{M}_u U_i$. Il est ensuite possible d'estimer $\widehat{\mathbf{M}}_u$ en résolvant un problème d'alignement 3d-3d [8]. Il est aussi possible de répéter la même approche de calibration avec des caméras virtuelles centrées sur les points clés de l'objet comme présenté Appendice A. Il y a donc trois stratégies possible :

Symétrique : estimer $\widehat{\mathbf{M}}_s$ avec des caméras virtuelles centrées sur l'utilisateur et $\widehat{\mathbf{M}}_u$ avec des caméras virtuelles centrées sur les points clés de l'objet.

Caméras centrées sur l'utilisateur seulement : estimer $\widehat{\mathbf{M}}_s$ avec des caméras virtuelles centrées sur l'utilisateur et $\widehat{\mathbf{M}}_u$ par alignement 3d-3d.

Caméras centrées sur les points clés seulement : estimer $\widehat{\mathbf{M}}_u$ avec des caméras virtuelles centrées sur les points clés de l'objet et $\widehat{\mathbf{M}}_s$ par alignement 3d-3d.

L'étape la plus délicate de la calibration est la DLT. Elle est sensible au bruit sur les observations et non sur la position du centre des caméras. Il est donc préférable de choisir comme centres les données les plus bruitées (positions utilisateurs ou points clés de l'objet). En pratique, il est plus difficile de suivre l'utilisateur que de localiser un objet connu. Le bon conditionnement de la DLT est également lié à la contrainte de non-planarité des points de calibrations (voir Section 2.6), difficilement applicable pour les positions utilisateurs limitées par le champ de vision à travers l'écran. Pour obtenir la meilleure estimation possible indépendamment de la configuration du problème, l'erreur de reprojection est estimée avec chaque approche et le meilleur résultat est conservé.

2.5 Ajustement de faisceaux

Après avoir estimé $\widehat{\mathbf{M}}_u$ et $\widehat{\mathbf{M}}_s$, un ajustement de faisceaux global avec covariances minimise la fonction (6) en utilisant l'algorithme de Levenberg-Marquardt.

2.6 Discussions

Contraintes et cas dégénérés Les contraintes sur la géométrie du problème viennent de l'étape DLT. On a déjà établi la contrainte $m \geq 6$ pour obtenir une solution unique. De plus, certaines configurations des points clés peuvent amener à des cas dégénérés. Ces cas sont traités dans [7], le plus notable étant le cas où la caméra et les points appartiennent à l'union d'un plan et d'une ligne droite passant par le centre de la caméra. L'objet de référence ne peut donc pas être plat ni trop éloigné de $\mathcal{C}_s$.

Optimisation multi-vue Le problème de l'agrégation des résultats (Section 2.3) provient du fait qu'aucune contrainte multi-vue n'est utilisée. Si on formule le problème avec une matrice d'observation on obtient :

$$\begin{bmatrix} \lambda_{11}c_{11} & \cdots & \lambda_{1n}c_{1n} \\ \vdots & & \vdots \\ \lambda_{m1}c_{m1} & \cdots & \lambda_{mn}c_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_1[\mathbf{R}_s|T_s - T_1] \\ \vdots \\ \mathbf{K}_m[\mathbf{R}_s|T_s - T_m] \end{bmatrix} \begin{bmatrix} O_1 \cdots O_n \end{bmatrix} \tag{16}$$

où les λ_{ij} sont les profondeurs projectives, calculables à partir des matrices fondamentales ou par des méthodes itératives [11, 18]. On observe que l'approche DLT ignore trois contraintes : le fait que $\mathbf{R}_s$ et T_s sont partagés par toutes les vues et que les T_i sont liés aux $\mathbf{K}_i$ (8). À notre connaissance, il n'y a aucun moyen de calculer une estimation directe en utilisant ces contraintes. De plus notre approche a l'avantage de rendre le processus aisément parallélisable.

Généralisation de la méthode La méthode a été introduite dans le cadre d'un système particulier mais nous nous sommes efforcé de ne pas utiliser d'information a priori. Cela permet plusieurs généralisations : des systèmes de type lunettes peuvent être calibrés en utilisant une seule caméra virtuelle centrée sur l'utilisateur. À l'opposé, en utilisant un seul point clé il est possible de calibrer un système de suivi de tête. Une configuration intéressante est la *vitrine augmentée* où le système est fixe dans la scène. Il n'y

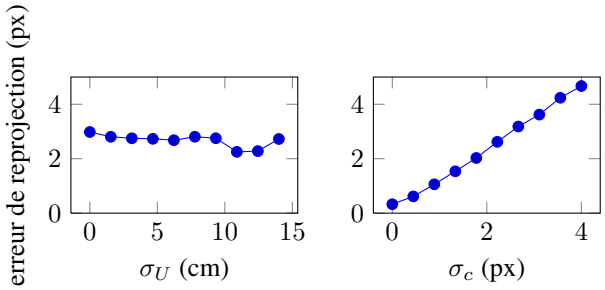


FIGURE 4 – Ajustement de faisceaux sur données synthétiques

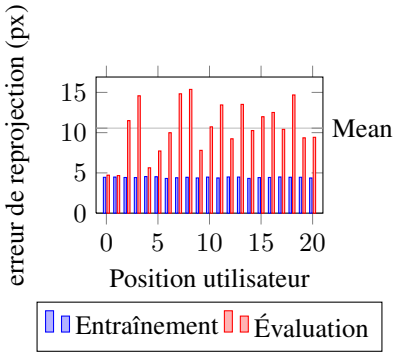


FIGURE 5 – Validation croisée

a dans ce cas pas besoin d'une caméra $\mathcal{C}_s$ dont le repère est remplacé par le repère local de l'objet : $\mathbf{M}_s$ est alors la pose de l'objet dans $\mathcal{W}$. Il s'agit de la configuration choisie pour l'évaluation (Appendice B) car elle permet de s'affranchir du calcul de la pose de l'objet observé.

3 Évaluation

Cette section est dédiée à la démonstration de la robustesse de notre méthode et de sa précision. L'algorithme a été implémenté en Python et les temps de calculs sont compris entre 1 et 3 secondes par calibration.

Évaluation sur données synthétiques Pour étudier le comportement de l'initialisation convexe, une scène synthétique est générée avec une géométrie et des niveaux de bruit similaire au cas réel présenté Appendice B. À partir de cette référence réaliste, l'influence des différents paramètres est mesurée 50 fois avant de calculer la moyenne et l'écart-type de l'erreur. L'erreur en rotation et translation est observée plutôt que l'erreur de reprojection dans les caméras virtuelles qui n'est pas un bon indicateur de la qualité de l'initialisation. Figure 3, lignes 1 et 2, on peut voir que le bruit sur les positions utilisateur n'affecte pas les caméras centrées sur ces dernières, et le bruit sur les points clés n'affecte pas les caméras centrées sur les points clés. Le résultat le plus intéressant de ces expériences est l'importance cruciale de la géométrie du problème (voir Section 2.6), révélée par la courbe sur la taille de l'objet (ligne 3).
La précision de l'ajustement de faisceau est évaluée Figure 4. L'erreur de reprojection est presque constante à 3 pixels, l'écart-type du bruit. La solution est donc optimale.

Test sur données réelles Le processus de calibration est mis à l'épreuve dans une configuration réelle décrite en B. Un affichage en fil de fer de la boîte est affiché sur l'écran transparent. L'erreur d'alignement est presque imperceptible ce qui montre la pertinence de l'approche. Des résultats quantitatifs sont produits au moyen d'une validation croisée sur 20 positions utilisateurs. Les poses de la caméra et de l'objet sont estimées en utilisant 19 positions et l'erreur d'alignement est calculée du point de vue de la position restante. Les résultats sont répertoriés Figure 5. L'erreur de reprojection moyenne est d'environ 10 pixels ce qui correspond à moins de 3 mm à l'écran. Des exemples de reprojections sont visibles Figure 6.

Comparaison avec travaux similaires Il est intéressant de comparer ce résultat aux méthodes utilisant un miroir [13, 16]. Ces deux publications présentent des résultats en conditions réelles comparables et rapportent une erreur de $7^\circ = 0.122\,\mathrm{rad}$ dans [13] et $0.057\,\mathrm{rad}$ dans [16]. Pour de faibles erreurs sur l'orientation d'une caméra, l'erreur sur la position de l'utilisateur ou de l'objet est $\alpha \cdot d$ where d où d est la distance à la caméra.
Pour un utilisateur à 70 centimètres de la caméra (position moyenne dans notre configuration), une erreur de $0.06\,\mathrm{rad}$ se traduit par une erreur de $0.06 \times 0.7 = 0.04$ mètres. Pour un point clé à 1 mètre de l'écran, l'erreur d'alignement qui en résulte est approximativement $\frac{1\times 0.04}{1+0.7} \approx 0.024$ mètres, sans prendre en compte l'erreur en translation et les imperfections des algorithmes de localisation. Cette borne inférieure de l'erreur, plus de 7 fois supérieure à nos résultats, est trop importante pour des applications de réalité augmentée.

Conclusion

Cet article présente une solution à la calibration des systèmes de réalité augmentée utilisant un affichage semi-transparent et deux caméras. Notre contribution principale consiste en une initialisation convexe grâce à l'introduction de caméras virtuelles suivie d'un ajustement de faisceaux. Les expériences ont démontré la précision et la robustesse de l'approche, ainsi que sa généralisation à des configurations plus communes. Nous envisageons par la suite d'intégrer un suivi utilisateur plus performant, et une chaîne complète de réalité augmentée. Il est également envisageable d'adapter la méthode à des surfaces non-planes telles que les pare-brise de voiture.

Appendice

A Caméras virtuelles centrées sur les points clés

Il est possible d'effectuer la calibration de caméras virtuelles (voir Section 2.2) centrées sur les points clés de

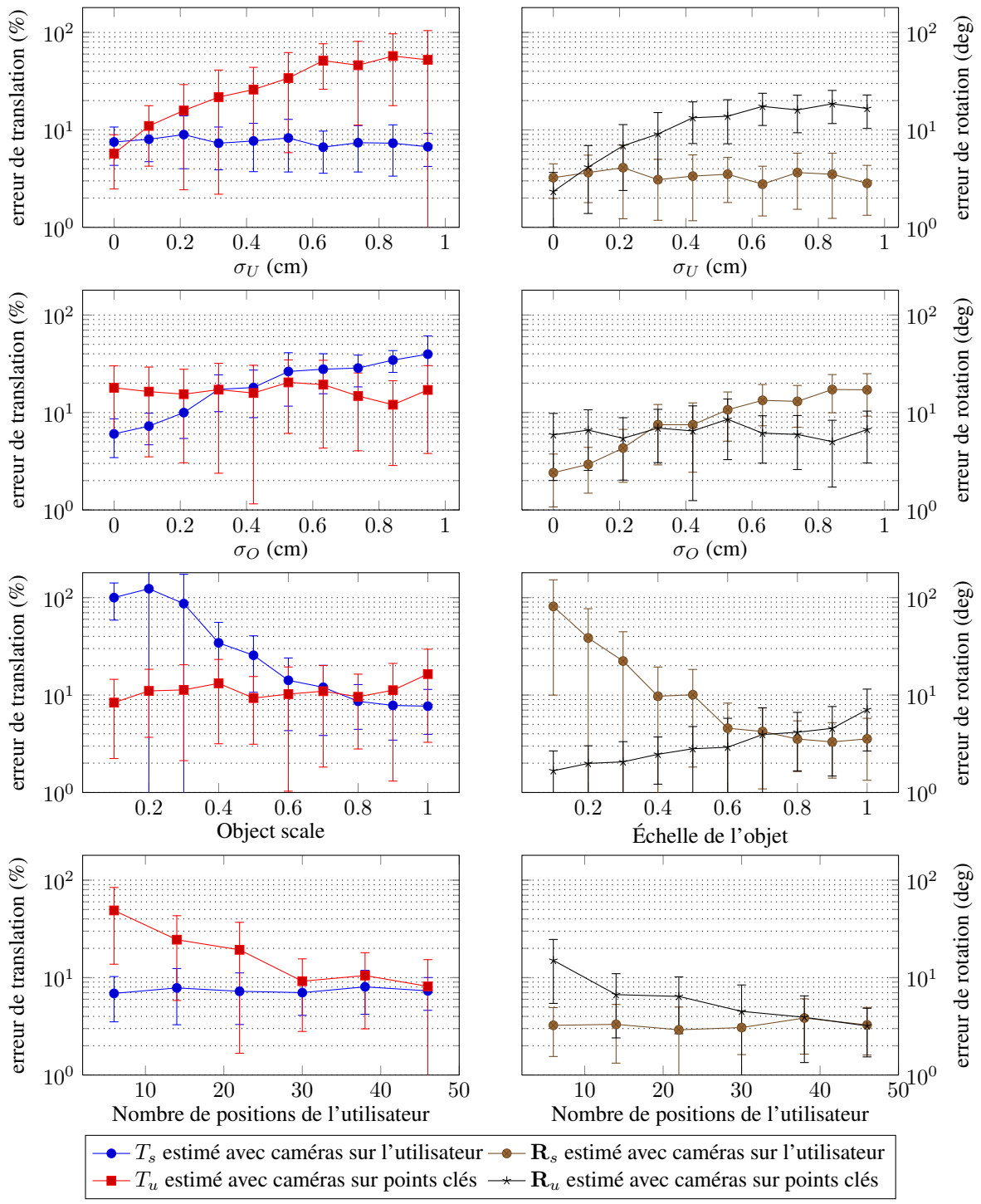


FIGURE 3 – Comparaison de la robustesse au bruit sur les positions utilisateur, position des points clés, échelle de l'objet et nombre de positions utilisateur. Les points correspondent à a valeur moyenne sur 50 échantillons et les barres à l'écart-type. L'échelle en y est logarithmique.

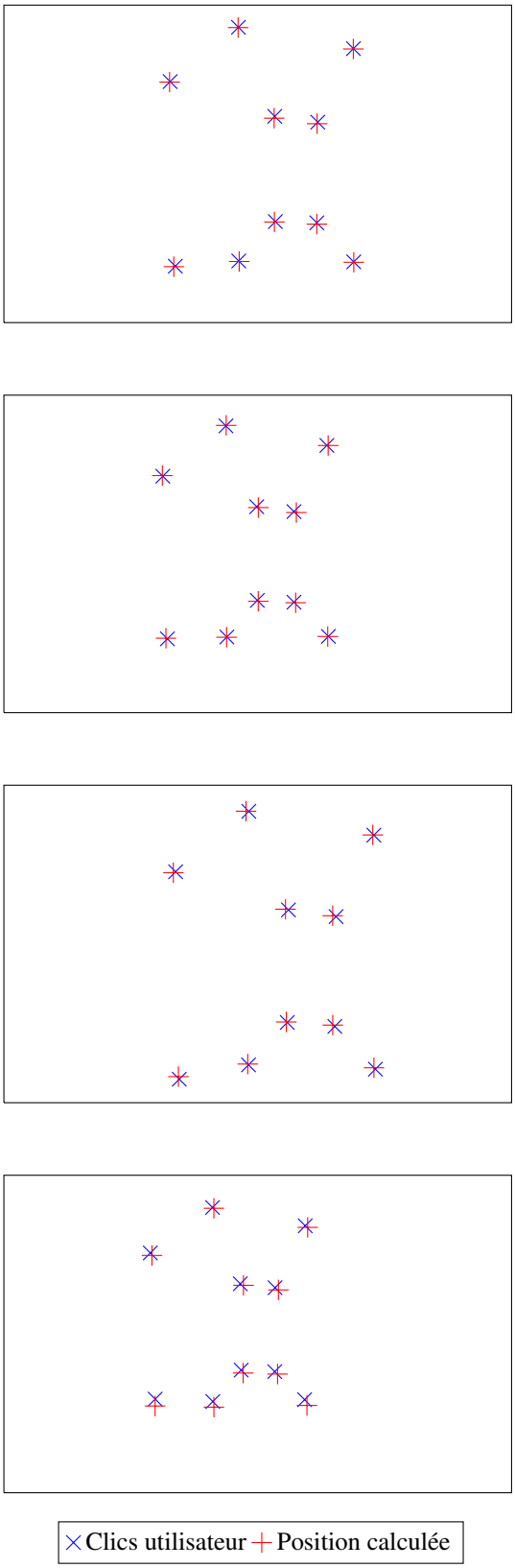


FIGURE 6 – Exemples d'alignement sur les données de validation croisée.

l'objet. Le problème est symétrique avec quelques changements. Comme leurs axes optiques sont dans la direction $-z$, les poses des caméras virtuelles sont définies par $\mathbf{M}_j^O = [\mathbf{R}^O|T_j^O]$ où $\mathbf{R}^O = \text{diag}(-1, 1, -1)$ et $T_j^O = \mathbf{M}_s O_j$. L'équation (8) devient :

$$\mathbf{K}_j^O = \begin{bmatrix} (T_j^O)_z & 0 & -(T_j^O)_x \\ 0 & (T_j^O)_z & (T_j^O)_y \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

et les projections des points $U_{i=1\ldots m}$ dans la caméra virtuelle j sont $c_{ij}^O = (-(c_{ij})_x, (c_{ij})_y)^T$. Le reste du processus de résolution est identique et permet d'obtenir une estimation $\widehat{\mathbf{M}}_u$ de la pose de $\mathcal{C}_u$.

B Détails du système et estimation du bruit

Les étapes de raffinement non linéaire et d'ajustement de faisceaux nécessitent une estimation de la variance du bruit sur les positions de l'utilisateur $\mathbf{\Sigma}_U$, les points clés de l'objet $\mathbf{\Sigma}_O$ et les clics de l'utilisateur $\mathbf{\Sigma}_c$. On définit ici une configuration réelle pour l'évaluation.

Points clés de l'objet Comme l'objet est statique dans la configuration choisie, la caméra $\mathcal{C}_s$ est définie comme les coordonnées locales de l'objet (configuration *vitrine augmentée* Section 2.6). Le bruit sur les points clés est donc uniquement imputable aux erreurs du modèle 3d. En pratique ces dernières sont très faibles, on établit

$$\mathbf{\Sigma}_O = \text{diag}(1^2, 1^2, 1^2) \text{ en millimètres} \quad (18)$$

L'objet est une boîte rectangulaire de dimensions $372 \times 305 \times 229$ mm et est placé à environ 1 mètre de l'écran. 10 points clés sont choisis : les 6 coins visibles ainsi que 4 points additionnels sur les côtés.

Positions de l'utilisateur Il n'y a aucune contrainte sur la méthode de suivi utilisée pour l'utilisateur. Il est possible d'utiliser une solution commerciale monoculaire telle que [14]. Nous utilisons une caméra stéréo USB Minoru [2] ($2 \times 640 \times 480$). La pupille gauche de l'utilisateur est détectée dans chaque caméra [19], puis un algorithme Mean-Shift [5] permet de trouver le centre de la pupille. Le principal avantage de cette méthode est d'être basée détection et donc plus robuste que les méthodes nécessitant un suivi dans le temps. Cependant, il est difficile de localiser précisément la pupille avec une telle caméra et la position obtenue est donc bruitée avec de nombreux sauts. Pour modéliser le bruit, nous avons observé le comportement de l'estimation pour un utilisateur immobile pendant plusieurs centaines d'images. La variance observée est :

$$\mathbf{\Sigma}_U = \text{diag}(5^2, 5^2, 20^2) \text{ en millimètres} \quad (19)$$

L'utilisateur doit rester immobile pour chaque série de clics. Sa position est mesurée à chaque clic et la moyenne

2. http://www.minoru3d.com/

est calculée. Pour une séquence de n clics, la variance de la position utilisateur est donc divisée par n : $\mathbf{\Sigma}_{U_{\text{average}}} = \frac{1}{n}\mathbf{\Sigma}_U$. Nous avons 10 clics par séquence donc :

$$\mathbf{\Sigma}_{U_{\text{average}}}^{(n=10)} = \text{diag}(1.6^2, 1.6^2, 6.3^2) \text{ en millimètres} \quad (20)$$

La caméra a un champ de vision restreint qui empêche l'utilisateur de sortir de la zone où :

$$U_i \in [-300, 300] \times [-150, 150] \times [400, 1000] \text{ en mm} \quad (21)$$

Cette zone est réduite par la contrainte que tous les points clés doivent être visibles à travers l'écran. Le nombre de points de vue différents pour l'évaluation est fixé à 20.

Clics utilisateur L'écran transparent est un écran SAMSUNG 22 pouces LTI220MT02. La surface active est de 473.6×296.1 mm avec une résolution de 1680×1050 pixels. La taille d'un pixel est donc $s_{\text{px}} = 0.282$ mm/px.
Le *bruit* des clics utilisateur a plusieurs origines : flou introduit par l'écran transparent, précision de la souris (un pixel), forme du curseur, tremblements de la main, défauts de vision, impossibilité de rester complètement immobile et autres facteurs humains. Dans la configuration choisie, une étude utilisateurs a montré que l'erreur moyenne est d'environ 3 pixels.

$$\begin{aligned} \mathbf{\Sigma}_c &= \text{diag}(3^2, 3^2) && \text{en pixels} \\ &= \text{diag}(0.846^2, 0.846^2) && \text{en millimètres} \end{aligned} \quad (22)$$

Références

[1] Y. ABDEL-AZIZ et H. KARARA. "Direct linear transformation from comparator to object space coordinates in close-range photogrammetry". Dans : *ASP Symposium on Close-Range Photogrammetry*. 1971.

[2] N ANJUM, M TAJ et A CAVALLARO. "Relative Position Estimation of Non-Overlapping Cameras". Dans : *ICASSP*. T. 2. 2007.

[3] J. BRAUX-ZIN et al. "Calibrating an Optical See-Through Rig with Two Non-overlapping Cameras : The Virtual Camera Framework". Dans : *3DIMPVT*. 2012.

[4] Y CASPI. "Alignment of non-overlapping sequences". Dans : *ICCV* (2001).

[5] D COMANICIU, V RAMESH et P MEER. "Real-time tracking of non-rigid objects using mean shift". Dans : *CVPR*. T. 2. 2000.

[6] Sandro ESQUIVEL, Felix WOELK et Reinhard KOCH. "Calibration of a Multi-camera Rig from Non-overlapping Views". Dans : *Pattern Recognition*. Sous la dir. de Fred HAMPRECHT, Christoph SCHNÖRR et Bernd JÄHNE. T. 4713. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007.

[7] R.˜I. HARTLEY et A ZISSERMAN. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, 2004.

[8] BKP HORN. "Closed-form solution of absolute orientation using unit quaternions". Dans : *Journal of the Optical Society of America A* 4.April (1987).

[9] RK KUMAR et al. "Simple calibration of non-overlapping cameras with a mirror". Dans : *CVPR*. 2008.

[10] Pierre LEBRALY et al. "Fast calibration of embedded non-overlapping cameras". Dans : *ICRA*. IEEE, mai 2011.

[11] John OLIENSIS et Richard HARTLEY. "Iterative extensions of the sturm/triggs algorithm : Convergence and nonconvergence". eng. Dans : *IEEE transactions on pattern analysis and machine intelligence* 29.12 (2005).

[12] A RAHIMI, B DUNAGAN et T DARRELL. "Simultaneous calibration and tracking with a network of non-overlapping sensors". Dans : *CVPR*. 2004.

[13] Rui RODRIGUES, J. BARRETO et Urbano NUNES. "Camera pose estimation using images of planar mirror reflections". Dans : *ECCV 2010* (2010).

[14] SEEING MACHINES. *FaceApi*. URL : http://www.seeingmachines.com/product/faceapi/.

[15] Peter STURM et Thomas BONFORT. "How to Compute the Pose of an Object without a Direct View ?" Dans : *ACCV*. T. 2. 2006.

[16] Kosuke TAKAHASHI, Shohei NOBUHARA et Takashi MATSUYAMA. "A New Mirror-based Extrinsic Camera Calibration Using an Orthogonality Constraint". Dans : *CVPR*. 2012.

[17] Arthur TANG, Ji ZHOU et Charles OWEN. "Evaluation of Calibration Procedures for Optical See-Through Head-Mounted Displays". Dans : *ISMAR*. Washington, DC, USA : IEEE Computer Society, 2003.

[18] Bill TRIGGS. "Factorization Methods for Projective Structure and Motion". Dans : *CVPR*. 1996.

[19] P VIOLA et M JONES. "Rapid Object Detection using a Boosted Cascade of Simple Features". Dans : *CVPR*. 2001.