Caméras virtuelles pour l'étalonnage d'un système de réalité augmentée sur affichage semi-transparent

Jim Braux-Zin [1], **Adrien Bartoli** [2], **Romain Dupont** [1], **Mohamed Tamaazousti** [1]

1. *CEA, LIST — 91191 Gif-sur-Yvette, France*
   jim.braux-zin@cea.fr
2. *ISIT, Université d'Auvergne — 63000 Clermont-Ferrand, France*
   adrien.bartoli@gmail.com

RÉSUMÉ. *Nous proposons dans cet article une nouvelle méthode d'étalonnage extrinsèque pour systèmes de réalité augmentée sur affichages semi-transparents. Cette méthode est appliquée ici à un système de type* tablette augmentée *composé d'un écran semi-transparent, d'un dispositif de suivi de l'utilisateur et d'un dispositif de localisation par rapport à la scène observée. Cette méthode reste cependant générique et applicable à la plupart des scénarios de réalité augmentée. Elle estime les poses relatives de ces trois composants en se basant sur les obervations 2D à l'écran de points de référence d'un objet connu. Ces observations sont fournies par l'utilisateur, par exemple sous forme de clics à la souris. Une initialisation convexe est d'abord calculée grâce à l'étalonnage de caméras virtuelles. Un ajustement de faisceaux global raffine ensuite le résultat. Des expériences sur données synthétiques et réelles montrent le bien-fondé de cette approche.*

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 user-tracking device and a device dedicated to the localization of the system in the environment but easily generalizable to any optical see-through setup. The proposed algorithm estimates the relative poses of those three component 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.*

MOTS-CLÉS : *étalonnage, calibrage, réalité augmentée, écran semi-transparent, caméras à champs disjoints.*

KEYWORDS: *calibration, augmented reality, optical see-through, non-overlapping cameras, semi-transparent screen.*

DOI:10.3166/TS.31.175-195

Extended abstract

Optical see-through augmented reality systems, using semi-transparent displays, offer compelling advantages with regard to standard video-see-through systems. The more important one is the ability to never isolate the user from reality. This is crucial for the critical applications we envision such as driving or surgery assistance, where lives are at stake. However, optical see-through systems are challenging to calibrate, ensuring proper alignment between the virtual augmentation and reality. Indeed one needs to compute the relative poses of the user, the observed scene and the semi-transparent screen for a correct display.

With no loss of generality, we consider a system made of a semi-transparent screen rigidly tied to two localization devices (one tracking the user and the other one tracking the observed scene). The calibration process involves estimating the pose of those two devices with regard to the screen and minimizing the alignment error between the augmentations and reality. To that end, the user is asked to indicate the on-screen projection of reference points of a known 3D object. The sought poses can be computed through non-linear optimization, minimizing the 2D distance between the user-provided projections and the ones computed by the system from the poses. However, without *a priori* knowledge about the geometry of the problem, the cost function is highly non-convex and difficult to optimize.

We propose in this article a new framework allowing to get a direct initial estimate of the poses by resectioning virtual cameras. Those virtual cameras are defined by their optical centers, coinciding with user positions, and their focal planes, all coinciding with the physical screen. The projections indicated by the user can then be seen as 2D observations of the 3D reference points in those cameras, allowing calibration by standard techniques. A similar reasoning allows us to define virtual cameras centered on the 3D reference points, "looking at" the user positions.

Synthetic and real experiments demonstrate the validity of this approach. Moreover, it is a generic method making no assumption on the system geometry, the type of localization devices (cameras, electromagnetic tracking…) or the display technology used (LCD screen, beam-splitter…).

1. Introduction

Les systèmes de réalité augmentée classiques (*video see-through*) superposent des éléments virtuels à un flux vidéo. Le faible coût du matériel nécessaire et la simplicité de la technique (voir figure 1, gauche) ont permis un essor important de cette technologie (Juniper Research, 2012). Cependant, ces approches n'affichent l'image qu'après l'avoir traitée, ce qui introduit une latence (acquisition, traitement et rendu graphique). De plus, les caméras actuelles ont une résolution, un taux de rafraîchissement et un contraste très limités par rapport aux capacités visuelles humaines. Enfin et surtout, de tels systèmes peuvent couper totalement l'utilisateur de la réalité en cas de défaillances matérielles ou logicielles.

Ces problèmes, peu gênants pour les applications ludiques, sont inacceptables pour les applications critiques telles que l'assistance aux opérations chirurgicales ou l'aide à la conduite, qui ne peuvent tolérer aucune indirection entre la réalité et l'utilisateur. Des systèmes utilisant un affichage semi-transparent sont plus adaptés : même si l'augmentation elle-même peut présenter des défauts, la réalité observée par transparence est toujours inaltérée. On parle alors d'*optical see-through*. Le principe de fonctionnement de tels systèmes est schématisé figure 1, montrant clairement la complexité ajoutée par rapport aux systèmes classiques *video see-through*. En particulier l'affichage dépend alors de la pose de la scène observée et de la position de l'utilisateur *par rapport à l'écran*. Jusqu'à présent, ces systèmes sont cependant restés limités à des usages de niche (tels que l'affichage tête haute dans les avions de chasse) car trop chers et encombrants. Par ailleurs, un intérêt industriel croissant s'affiche pour les lunettes semi-transparentes. Malgré des progrès indéniables, ces systèmes restent cependant trop intrusifs et sujets aux rapides mouvements de tête rendant l'analyse de la scène difficile.

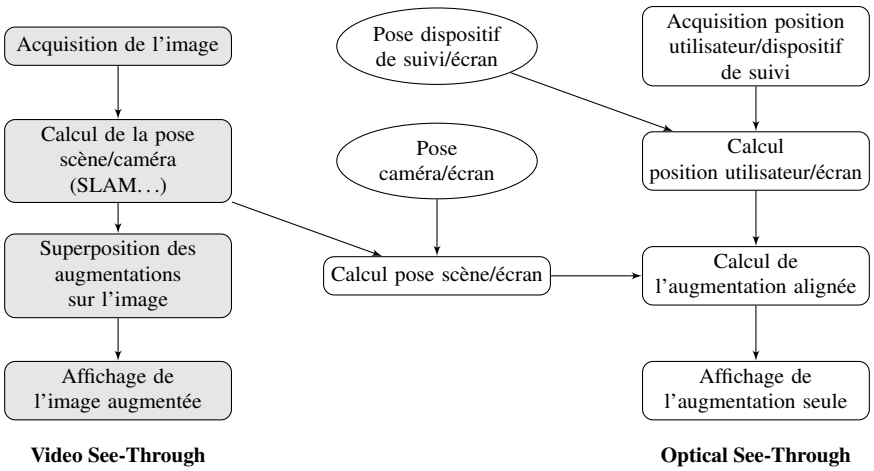


Figure 1. Étapes principales des processus video see-through *(en gris) et* optical see-through *(en blanc). On observe que ce dernier est beaucoup plus complexe et nécessite notamment les poses relatives des dispositifs de localisation par rapport à l'écran (représentées par des ovales). Celles-ci sont fixes pendant l'exécution*

Nous nous intéressons ici à une approche « tablette » utilisant un écran LCD semi-transparent sur lequel sont fixés un dispositif de suivi de l'utilisateur et un dispositif de localisation dans l'environnement (figure 2). Cela permet de s'affranchir de la plupart des inconvénients des systèmes cités précédemment : l'affichage semi-transparent est adapté aux applications critiques, le système est mobile, capable de se localiser dans l'environnement et n'est pas soumis à des mouvements aussi rapides que sur des lunettes. Cependant, les dispositifs opérant dans leurs repères respectifs, l'estimation de leurs poses par rapport à l'écran est nécessaire pour aligner l'affichage de l'information virtuelle avec la réalité (voir figure 1).

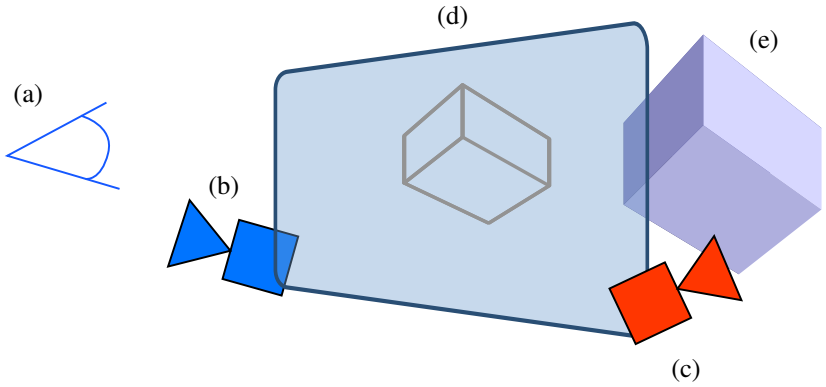
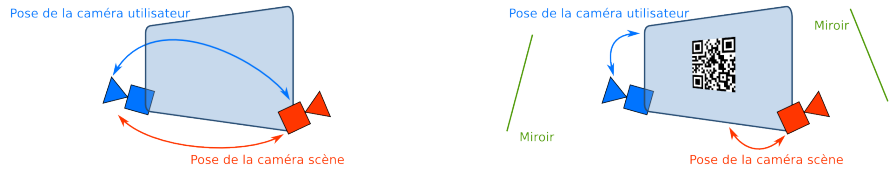


Figure 2. Représentation schématique du système : (a) utilisateur, (b) dispositif de suivi de l'utilisateur, (c) dispositif de localisation dans l'environnement, (d) écran semi-transparent, (e) scène

À notre connaissance, le problème de l'étalonnage extrinsèque d'un tel système n'est abordé dans la littérature que dans le cas où les dispositifs de localisation sont des caméras. Dans la configuration décrite en figure 2, les deux caméras ont des champs de vision disjoints et aucune ne voit l'écran. Les méthodes classiques d'étalonnage extrinsèque utilisant un motif connu (Zhang, 2000) sont donc inapplicables. D'autres méthodes ont des objectifs proches et peuvent être regroupées en deux familles (figure 3). D'abord l'étalonnage extrinsèque de deux caméras à champs disjoints a été abondamment traité dans la littérature : plusieurs méthodes se basent sur la localisation des caméras mobiles (Caspi, Irani, 2002 ; Esquivel *et al.*, 2007 ; Lébraly *et al.*, 2011) ou statiques (Rahimi *et al.*, 2004 ; Anjum *et al.*, 2007). Ces méthodes permettent d'obtenir les poses relatives des deux caméras 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 (Sturm, Bonfort, 2006 ; Kumar *et al.*, 2008 ; Rodrigues *et al.*, 2010 ; Takahashi *et al.*, 2012). En appliquant ces méthodes pour estimer la pose de chaque caméra par rapport à l'écran, il est théoriquement possible d'étalonner complètement le système. Le premier inconvénient de ces méthodes est le fait que les deux caméras sont étalonnées totalement indépendamment, ce qui rend le processus plus lourd à mettre en œuvre et moins précis. De plus, ces méthodes minimisent une erreur de reprojection dans le plan image de la caméra. Or 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 (représentée sur la figure 4). Il s'agit en effet de l'objectif de la plupart des méthodes d'étalonnage des systèmes de type lunettes (Tang, 2003).

Nous proposons dans cet article une nouvelle méthode qui minimise directement l'erreur d'alignement grâce aux observations indiquées par l'utilisateur (figure 4). En considérant le système dans son ensemble l'approche est indépendante des capteurs et algorithmes utilisés pour le suivi. Il est par exemple possible d'utiliser une caméra



(a) Étalonnage extrinsèque de caméras à champs disjoints : la pose de l'écran est manquante

(b) Méthodes utilisant un miroir : deux étalonnages caméra/écran indépendants sont nécessaires

Figure 3. Illustration des méthodes d'étalonnage de l'état de l'art et leurs limites

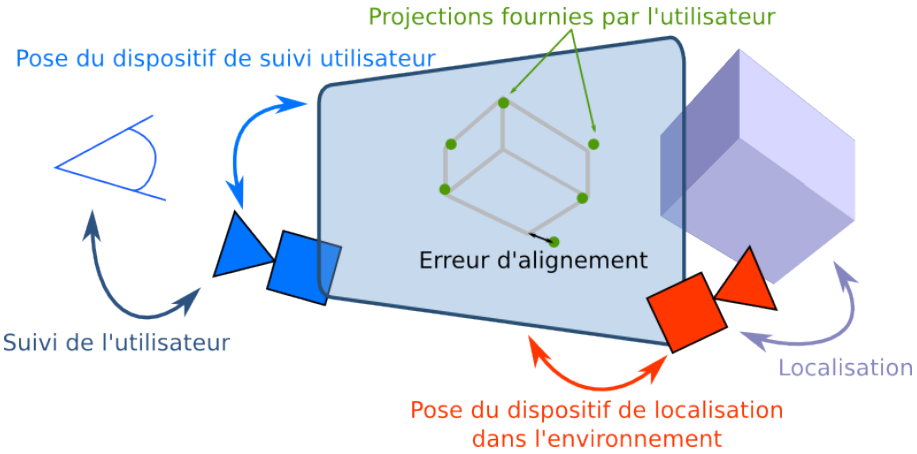


Figure 4. Représentation du problème d'étalonnage : les poses des deux dispositifs de localisation par rapport à l'écran sont les inconnues, l'erreur d'alignement est l'énergie à minimiser et les autres informations sont les données en entrées. A visualiser de préférence en couleur

stéréoscopique [1] 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 en partie compensées.

Énoncé du problème et plan

Le processus d'étalonnage, formalisé en section 2, doit minimiser l'erreur de reprojection perçue par l'utilisateur. La fonction de coût associée est non convexe. Pour

1. La solution retenue pour le prototype est une caméra stéréoscopique bas coût associée à un algorithme de détection de pupille, voir section 5.1.

garder la méthode générique [2], aucune hypothèse n'est faite sur les poses des dispositifs de localisation et l'initialisation est effectuée grâce à une approximation convexe robuste basée sur l'introduction de caméras virtuelles. Ce processus est expliqué en section 3 ainsi que l'étape d'ajustement de faisceaux qui suit. Enfin la section 5 est dédiée à l'évaluation de la précision de la méthode sur des données synthétiques et réelles.

2. Formulation du problème

2.1. Notations et conventions

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 $\mathbf{M}$. Les matrices diagonales sont représentés par leurs coefficients diagonaux sous la forme $\mathrm{diag}(x_1, x_2, \dots)$. La même notation sous forme de lettre calligraphiée $\mathcal{C}$ est utilisée pour désigner un dispositif de localisation et son repère associé. Le repère monde est noté $\mathcal{W}$. Un vecteur exprimé dans le repère $\mathcal{C}$ est notée $X^{(\mathcal{C})}$. Nous utiliserons la norme de Mahalanobis notée $||x||_\Sigma = \sqrt{x^T \Sigma^{-1} x}$.

Le système visible sur la figure 2 est composé de trois composants fixés rigidement : l'écran semi-transparent, le dispositif de suivi utilisateur $\mathcal{C}_u$ et et le dispositif de localisation dans l'environnement $\mathcal{C}_o$. L'écran transparent est utilisé comme référence et définit le repère monde $\mathcal{W}$ où les axes sont illustrés en figure 5 et où l'origine est un point arbitraire de l'écran. La pose de $\mathcal{C}_u$ dans $\mathcal{W}$ s'exprime par la transformation $\mathbf{M}_{\mathcal{C}_u \to \mathcal{W}} = \left[\mathbf{R}_{\mathcal{C}_u \to \mathcal{W}} \quad | \quad T_{\mathcal{C}_u \to \mathcal{W}}\right]$ où $\mathbf{R}_{\mathcal{C}_u \to \mathcal{W}}$ est la matrice de rotation 3×3 et $T_{\mathcal{C}_u \to \mathcal{W}}$ le vecteur translation 3×1. De même, $\mathbf{M}_{\mathcal{C}_o \to \mathcal{W}}$, $\mathbf{R}_{\mathcal{C}_o \to \mathcal{W}}$, et $T_{\mathcal{C}_o \to \mathcal{W}}$ décrivent la pose de $\mathcal{C}_o$ dans $\mathcal{W}$.

Soient n points 3D de référence choisis dans l'environnement dont les coordonnées $O^{(\mathcal{C}_o)}_{j=1\dots n}$ dans $\mathcal{C}_o$ sont connues. Depuis m positions distinctes, notées $U^{(\mathcal{C}_u)}_{i=1\dots m}$ dans $\mathcal{C}_u$, l'utilisateur indique la position des observations 2D c_{ij} de ces points de référence dans l'écran. Pour une position utilisateur $U_i^{(\mathcal{C}_u)}$ et un point de référence $O_j^{(\mathcal{C}_o)}$, c_{ij} est l'intersection du rayon $(U_i^{(\mathcal{W})}, O_j^{(\mathcal{W})})$ avec le plan $z = 0$ (l'écran) dans $\mathcal{W}$:

$$c_{ij} = f_{\text{int}}(U_i^{(\mathcal{W})}, O_j^{(\mathcal{W})}) = f_{\text{int}}(\mathbf{M}_{\mathcal{C}_u \to \mathcal{W}} U_i^{(\mathcal{C}_u)}, \mathbf{M}_{\mathcal{C}_o \to \mathcal{W}} O_j^{(\mathcal{C}_o)}) \tag{1}$$

où f_{int} est la fonction d'intersection définie pour $z_1 > 0$ et $z_2 < 0$ par :

$$f_{\text{int}}\left(\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}\right) = \begin{pmatrix} x_1 - z_1 \times \frac{x_2 - x_1}{z_2 - z_1} \\ y_1 - z_1 \times \frac{y_2 - y_1}{z_2 - z_1} \end{pmatrix}. \tag{2}$$

2. Plusieurs généralisations intéressantes sont présentées en section 4.

2.2. Modèle de bruit

En pratique, l'estimation des positions de l'utilisateur et des points 3D de référence est sujette à du bruit. Les observations 2D fournies par l'utilisateur sont également imprécises. En effet quelle que soit la méthode employée, ces observations restent soumises à des facteurs technologiques (flou introduit par l'écran, précision du dispositif d'acquisition...) et humains (défauts de vision, tremblements...). Ces bruits sont modélisés par des variables gaussiennes. Les versions bruitées des variables d'entrée du système sont notées surmontées d'un tilde. Soit $\mathcal{N}(\mu, \mathbf{\Sigma})$ une variable aléatoire suivant la loi normale de moyenne μ et de matrice de covariance $\mathbf{\Sigma}$:

$$\widetilde{U}_i^{(\mathcal{C}_u)} = U_i^{(\mathcal{C}_u)} + \mathcal{N}(0_3, \mathbf{\Sigma}_U) \tag{3}$$

$$\widetilde{O}_j^{(\mathcal{C}_o)} = O_j^{(\mathcal{C}_o)} + \mathcal{N}(0_3, \mathbf{\Sigma}_O) \tag{4}$$

$$\widetilde{c}_{ij} = c_{ij} + \mathcal{N}(0_2, \mathbf{\Sigma}_c) \tag{5}$$

pour tout i entre 1 et m et j entre 1 et n. 0_2 et 0_3 sont les vecteurs nuls de dimensions respectives 2 et 3. Les bruits varient peu pour une configuration donnée du système et il est possible d'estimer leurs covariances (voir section 5.1 pour un exemple), et ainsi d'améliorer la précision de l'étalonnage. Il est nécessaire de distinguer les paramètres estimés de leurs véritables valeurs. Les estimations sont notées avec un chapeau : $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$, $\widehat{\mathbf{M}}_{\mathcal{C}_o \to \mathcal{W}}$, $\widehat{U}_{i=1\ldots m}^{(\mathcal{C}_u)}$, $\widehat{O}_{j=1\ldots n}^{(\mathcal{C}_o)}$.

2.3. Problème d'étalonnage

L'objectif de la méthode proposée est de trouver les meilleures estimations $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$ et $\widehat{\mathbf{M}}_{\mathcal{C}_o \to \mathcal{W}}$ sachant $\widetilde{U}_{i=1\ldots m}^{(\mathcal{C}_u)}$, $\widetilde{O}_{j=1\ldots n}^{(\mathcal{C}_o)}$ et les $\widetilde{c}_{ij}$ correspondants, grâce à l'équation (1). En considérant des bruits gaussiens, la solution optimale minimise la fonction de coût suivante :

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

Il s'agit d'un ajustement de faisceaux (Triggs *et al.*, 2000). On reconnaît dans le premier terme l'erreur d'alignement 2D, non convexe, alors que les autres sont les erreurs 3D sur les positions des points de référence et de l'utilisateur. La section suivante présente une méthode originale d'initialisation convexe.

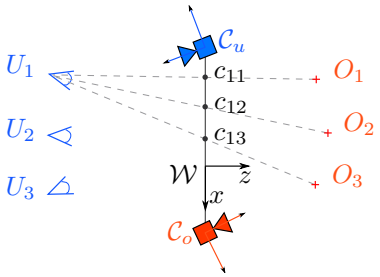


Figure 5. Notations (les éléments de même couleur sont exprimés dans le même repère)

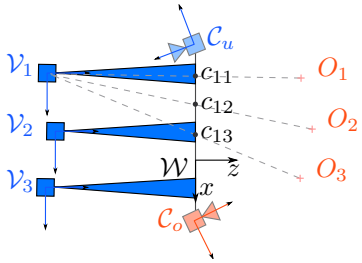


Figure 6. Caméras virtuelles utilisateur (voir texte)

3. Étalonnage et caméras virtuelles

3.1. Définition des caméras virtuelles

L'introduction de caméras virtuelles permet de décomposer le problème en plusieurs sous-problèmes classiques d'étalonnage. Soient 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 $\mathcal{W}$. Cette définition donne à ces caméras des propriétés intéressantes : tous leurs axes optiques sont parallèles et pointent dans la direction de l'axe z dans $\mathcal{W}$. Les poses des caméras virtuelles dans $\mathcal{W}$ sont donc définies par :

$$\mathbf{M}_i = \left[\mathbf{I} \quad | \quad U_i^{(\mathcal{W})}\right], \quad \forall i \in [\![1, m]\!] \tag{7}$$

où $\mathbf{I}$ est la matrice identité 3×3 et $U_i^{(\mathcal{W})} = \mathbf{M}_{\mathcal{C}_u \to \mathcal{W}} U_i^{(\mathcal{C}_u)}$ est la position $U_i^{(\mathcal{C}_u)}$ exprimée dans $\mathcal{W}$. Comme les observations c_{ij} sont exprimées en unités de $\mathcal{W}$, les «pixels» des caméras virtuelles sont parfaitement carrés : les longueurs focales en x et y sont égales ($f_{x_i} = f_{y_i} = f_i$), l'obliquité est nulle ($s_i = 0$) et il n'y a aucune distortion. De plus, comme les caméras sont définies par leur plan focal et point principal dans $\mathcal{W}$, leurs paramètres intrinsèques (point principal (u_i, v_i) et focale f_i) sont liés à la position de leur centre optique $U_i^{(\mathcal{W})} = \left(\left[U_i^{(\mathcal{W})}\right]_x, \left[U_i^{(\mathcal{W})}\right]_x, \left[U_i^{(\mathcal{W})}\right]_z\right)^T$. La matrice des paramètres intrinsèques s'écrit alors de la manière suivante :

$$\begin{aligned} \mathbf{K}_i &= \begin{bmatrix} f_{x_i} & s_i & u_i \\ 0 & f_{y_i} & v_i \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_i & 0 & u_i \\ 0 & f_i & v_i \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -\left[U_i^{(\mathcal{W})}\right]_z & 0 & \left[U_i^{(\mathcal{W})}\right]_x \\ 0 & -\left[U_i^{(\mathcal{W})}\right]_z & \left[U_i^{(\mathcal{W})}\right]_y \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{8}$$

Les observations 2D c_{ij} sont les projections des points 3D de référence dans $\mathcal{V}_i$, donc en coordonnées homogènes $\forall i \in [\![1, m]\!]$, $\forall j \in [\![1, n]\!]$, $\exists \lambda_{ij} \in \mathbb{R}$:

$$c_{ij} = \lambda_{ij} \mathbf{K}_i \mathbf{M}_i^{-1} \mathbf{M}_{\mathcal{C}_o \to \mathcal{W}} O_j^{(\mathcal{C}_o)} \tag{9}$$

$$c_{ij} = \lambda_{ij} \mathbf{H}_i O_j^{(\mathcal{C}_o)} \tag{10}$$

où

$$\mathbf{H}_i = \mathbf{K}_i \mathbf{M}'_i \tag{11}$$

$$\mathbf{M}'_i = \mathbf{M}_i^{-1} \mathbf{M}_{\mathcal{C}_o \to \mathcal{W}} = \left[\mathbf{R}_{\mathcal{C}_o \to \mathcal{W}} | T_{\mathcal{C}_o \to \mathcal{W}} - U_i^{(\mathcal{W})} \right]. \tag{12}$$

3.2. Étalonnage 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 $\mathbf{M}_{\mathcal{C}_o \to \mathcal{W}}$ du dispositif de localisation $\mathcal{C}_o$ dans l'environnement. Ce processus d'étalonnage est détaillé par Hartley et Zisserman (2004). Nous utilisons l'approche *Direct Linear Transform* (DLT) (Abdel-Aziz, Karara, 1971) pour estimer $\mathbf{H}_i$ à partir de l'équation (10). Le facteur scalaire λ_{ij} est éliminé grâce à un produit vectoriel :

$$c_{ij} = \lambda_{ij} \mathbf{H}_i O_j^{(\mathcal{C}_o)} \quad \Rightarrow \quad c_{ij} \wedge \mathbf{H}_i O_j^{(\mathcal{C}_o)} = 0 \qquad \forall i \in [\![1, m]\!],\ \forall j \in [\![1, n]\!] \tag{13}$$

Pour chaque position utilisateur $U_i^{(\mathcal{C}_u)}$, ces équations vectorielles donnent $2 \times n$ équations linéaires indépendantes sur les coefficients de $\mathbf{H}_i$. Avec $n \geq 6$ points de référence, il est possible de calculer une solution aux moindres carrés, convexe hors des cas dégénérés (voir section 3.6.1), pour les 12 coefficients avec une décomposition en valeurs singulières [3].

L'étape suivante est d'extraire les paramètres intrinsèques $\mathbf{K}_i$ et extrinsèques $\mathbf{M}'_i$ vérifiant l'équation (11) : $\mathbf{H}_i = \mathbf{K}_i \mathbf{M}'_i$. On part d'une décomposition directe avec l'équation

$$\overline{\mathbf{H}}_i \overline{\mathbf{H}}_i^T = \mathbf{K}_i \mathbf{R}_{\mathcal{C}_o \to \mathcal{W}} \mathbf{R}_{\mathcal{C}_o \to \mathcal{W}}^T \mathbf{K}_i^T = \mathbf{K}_i \mathbf{K}_i^T \tag{14}$$

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

$$\widehat{\mathbf{K}}_i^{\text{init}} = \begin{bmatrix} f_{x_i} & s_i & u_i \\ 0 & f_{y_i} & v_i \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

3. La solution obtenue minimise la distance algébrique et non la distance euclidienne mais constitue une bonne initialisation. Voir Hartley et Zisserman (2004).

Cette décomposition initiale est raffinée itérativement en utilisant l'algorithme de Levenberg-Marquardt (Marquardt, 1963) pour minimiser :

$$f(\widehat{\mathbf{K}}_i, \widehat{\mathbf{M}}'_i, \widehat{O}^{(\mathcal{C}_o)}_{j=1\ldots n}) = \sum_{j=1\ldots n} \left\| \widetilde{c}_{ij} - \lambda_{ij} \widehat{\mathbf{K}}_i \widehat{\mathbf{M}}'_i \widehat{O}^{(\mathcal{C}_o)}_j \right\|^2_{\mathbf{\Sigma}_c} + \sum_{j=1\ldots n} \left\| \widetilde{O}^{(\mathcal{C}_o)}_j - \widehat{O}^{(\mathcal{C}_o)}_j \right\|^2_{\mathbf{\Sigma}_O} + w \left(|f_{x_i} - f_{y_i}| + |s_i| \right) \quad (16)$$

où w est un poids encourageant le matrice à prendre la forme désirée ($f_{x_i} = f_{y_i}$ et $s_i = 0$) et qui doit d'après Hartley et Zisserman (2004) 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 indépendant du niveau de bruit.

3.3. Extraction de la pose du dispositif $\mathcal{C}_o$

Il est maintenant possible d'extraire $\widehat{U}^{(\mathcal{W})}_i$ à partir de $\widehat{\mathbf{K}}_i$ grâce à l'équation (8). Nous construisons ensuite $\widehat{\mathbf{M}}_i$ à partir de $\widehat{U}^{(\mathcal{W})}_i$ en utilisant l'équation (7). Enfin $\widehat{\mathbf{M}}^{(i)}_{\mathcal{C}_o \to \mathcal{W}}$ est extrait de $\widehat{\mathbf{M}}'_i$ avec l'équation (12). Notons que ces estimations sont effectuées indépendamment pour chaque caméra virtuelle, d'où la présence de l'indice (i).

Toutes ces estimations $\widehat{\mathbf{M}}^{(i)}_{\mathcal{C}_o \to \mathcal{W}}$ doivent être agrégées en un seul $\widehat{\mathbf{M}}_{\mathcal{C}_o \to \mathcal{W}}$ pour initialiser l'ajustement de faisceaux. Dans un premier temps les échecs manifestes sont éliminés : caméra virtuelles pour lesquelles le raffinement non linéaire n'a pas convergé ou 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. Le lecteur intéressé pourra se référer à la section 3.6.2 pour une discussion à propos d'une solution optimale.

3.4. Extraction de la pose du dispositif $\mathcal{C}_u$

Les caméras virtuelles sont centrées sur les positions de l'utilisateur et l'étalonnage de chaque $\mathcal{V}_i$ donne une estimation $\widehat{U}^{(\mathcal{W})}_i$ de $U^{(\mathcal{W})}_i = \mathbf{M}_{\mathcal{C}_u \to \mathcal{W}} U^{(\mathcal{C}_u)}_i$. Il est ensuite possible d'estimer $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$ en résolvant un problème d'alignement 3D-3D (Horn, 1987). Il est aussi possible de répéter la même approche d'étalonnage avec des caméras virtuelles centrées sur les points 3D de référence de l'environnement comme présenté en Annexe A. Nous proposons donc trois stratégies :

Symétrique : estimer $\widehat{\mathbf{M}}_{\mathcal{C}_o \to \mathcal{W}}$ avec des caméras virtuelles centrées sur l'utilisateur et $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$ avec des caméras virtuelles centrées sur les points 3D de référence de l'environnement.

Caméras centrées sur l'utilisateur seulement : estimer $\widehat{\mathbf{M}}_{\mathcal{C}_o \to \mathcal{W}}$ avec des caméras virtuelles centrées sur l'utilisateur et $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$ par alignement 3D-3D.

Caméras centrées sur les points de référence seulement : estimer $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$ avec des caméras virtuelles centrées sur les points 3D de référence de l'environnement et $\widehat{\mathbf{M}}_{\mathcal{C}_o \to \mathcal{W}}$ par alignement 3D-3D.

L'étape la plus délicate de l'étalonnage est la DLT. Elle est sensible au bruit sur les observations et non sur la position du centre des caméras virtuelles. Il est donc préférable de choisir comme centres les données les plus bruitées (positions utilisateurs ou points 3D de référence de l'environnement). 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 d'étalonnage (voir la section 3.6.1), difficilement applicable pour les positions utilisateurs limitées par le champ de vision à travers l'écran. Ces deux arguments montrent que centrer les caméras virtuelles sur l'utilisateur seulement est *a priori* la meilleure stratégie. Cependant 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.

3.5. Ajustement de faisceaux

Les estimations $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$ et $\widehat{\mathbf{M}}_{\mathcal{C}_o \to \mathcal{W}}$ et les mesures bruitées $\widetilde{U}^{(\mathcal{C}_u)}_{i=1\ldots m}$ et $\widetilde{O}^{(\mathcal{C}_o)}_{j=1\ldots n}$ permettent d'initialiser l'ajustement de faisceaux global avec covariances qui minimise la fonction (6) en utilisant l'algorithme de Levenberg-Marquardt (Marquardt, 1963). Si l'algorithme converge (initialisation proche de la solution) et que les covariances des bruits sont correctement estimées, la solution obtenue est optimale du point de vue de l'utilisateur : l'erreur d'alignement est minimisée.

3.6. Discussions

3.6.1. Contraintes et cas dégénérés

Les contraintes géométriques viennent de l'étape DLT (section 3.2). Nous avons déjà établi que le nombre de points 3D de référence devaient respecter la contrainte $m \geq 6$ pour obtenir une solution unique. De plus, certaines configurations des points 3D de référence peuvent amener à des cas dégénérés, traités par Hartley et Zisserman (2004). Le cas le plus notable est le cas où les points sont coplanaires. L'objet de référence ne peut donc en particulier pas être plat ni trop éloigné de $\mathcal{C}_o$.

3.6.2. Optimisation multi-vue

Le problème de l'agrégation des résultats (section 3.3) provient du fait qu'aucune contrainte multi-vue n'est utilisée. Le problème (10) peut être reformulé avec une matrice d'observation de la manière suivante :

$$\begin{bmatrix} \lambda_{11}c_{11} & \cdots & \lambda_{1n}c_{1n} \\ \vdots & & \vdots \\ \lambda_{m1}c_{m1} & \cdots & \lambda_{mn}c_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_1[\mathbf{R}_{\mathcal{C}_o\to\mathcal{W}}|T_{\mathcal{C}_o\to\mathcal{W}} - T_1] \\ \vdots \\ \mathbf{K}_m[\mathbf{R}_{\mathcal{C}_o\to\mathcal{W}}|T_{\mathcal{C}_o\to\mathcal{W}} - T_m] \end{bmatrix} \begin{bmatrix} O_1^{(\mathcal{C}_o)} \cdots O_n^{(\mathcal{C}_o)} \end{bmatrix} \tag{17}$$

où les λ_{ij} sont les profondeurs projectives, calculables à partir des matrices fondamentales ou par des méthodes itératives (Triggs, 1996 ; Oliensis, Hartley, 2005). Dans notre contexte, nous observons que l'approche DLT ignore trois contraintes : le fait que $\mathbf{R}_{\mathcal{C}_o\to\mathcal{W}}$ et $T_{\mathcal{C}_o\to\mathcal{W}}$ sont partagés par toutes les vues et que les $U_i^{(\mathcal{W})}$ sont liés aux $\mathbf{K}_i$ par l'équation (8). À notre connaissance, il n'y a aucun moyen de calculer une estimation directe en utilisant ces contraintes. Notons par ailleurs que notre approche a l'avantage de rendre le processus aisément parallélisable pour une résolution plus rapide.

4. Scénarios envisageables et applications

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*, ce qui permet plusieurs généralisations. Les applications envisagées peuvent être classées en trois catégories.

4.1. Lunette augmentée

Dans le cas des lunettes augmentées ou *Head-Mounted Displays* (HMD), l'écran est rigidement fixé à l'utilisateur.

Avantages Il n'y a pas besoin de suivi de l'utilisateur $\mathcal{C}_u$. Il est aisé d'afficher un flux vidéo adapté à chaque œil pour que l'affichage soit aligné pour les deux yeux. Le système est mobile et ne restreint pas les mouvements de l'utilisateur.

Inconvénients La localisation par vision dans l'environnement est complexe à cause des mouvements rapides de rotation de la tête et des contraintes fortes sur la latence. Les solutions éprouvées sont basées sur une localisation externe (systèmes électromagnétiques, capture du mouvement avec marqueurs…). En l'état actuel des technologies, les HMD induisent une fatigue due aux incohérences perçues par le cerveau (principalement la latence).

Étalonnage La position utilisateur $U^{(\mathcal{W})}$ est constante. L'étalonnage peut être effectué en utilisant une seule caméra virtuelle centrée sur l'utilisateur. Ce dernier devra connecter un matériel spécifique (souris…) pour entrer la position des observations 2D des points 3D de référence. Alternativement, les points de référence peuvent être organisés en un motif spécifique et l'utilisateur devra se

déplacer pour faire coïncider le motif s'affichant sur son écran avec le motif réel ; cela permet d'effectuer l'étalonnage en une seule étape.

4.2. Vitrine augmentée

Dans la configuration de *vitrine augmentée*, le système est fixe dans la scène.

Avantages Il n'y a dans ce cas pas besoin du dispositif de localisation dans l'environnement $\mathcal{C}_o$. L'étalonnage fournit toutes les informations requises pour ajouter des informations en réalité augmentée sur l'objet de référence. De plus, en restreignant la réalité augmentée à une fenêtre bien définie dans l'espace, l'inconfort pour l'utilisateur est réduit car une grande partie de son champ de vision n'est pas affecté.

Inconvénients En l'absense de dispositif de localisation dans l'environnement, l'étalonnage doit être actualisé à chaque déplacement du système ou modification de la scène.

Étalonnage Le repère $\mathcal{C}_o$ est remplacé par le repère local de l'objet. L'étalonnage fournit la pose $\mathbf{M}_{\mathcal{C}_u \to \mathcal{W}}$ permettant de localiser l'utilisateur par rapport à l'écran ainsi que la pose $\mathbf{M}_{\mathcal{C}_o \to \mathcal{W}}$ qui est ici directement la pose de l'objet dans $\mathcal{W}$.

Il s'agit de la configuration choisie pour l'évaluation présentée en section 5.3 car elle permet de s'affranchir d'un algorithme de localisation dans l'environnement.

4.3. Tablette Augmentée

Les vitrines virtuelles sont limitées aux systèmes statiques, et en l'état actuel des technologies les HMD restent limités et trop intrusifs. C'est pourquoi nous croyons au paradigme de la *tablette augmentée*, utilisé tout au long de cet article. Cette approche utilise un écran semi-transparent, un dispositif de localisation dans l'environnement (pour que le système puisse être déplacé sans nouvel étalonnage) et un dispositif de suivi de l'utilisateur (pour que l'utilisateur soit libre de ses mouvements par rapport au système). Un tel système combine les avantages des deux approches précédentes. Il existe des solutions pour garder libres les mains de l'utilisateur (les applications visées sont l'aide à la chirurgie et à la maintenance de systèmes complexes). Par exemple un bras articulé, déjà courant dans les environnements médicaux, pourrait maintenir la tablette en place.

5. 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 calcul sont compris entre 1 et 3 secondes par étalonnage.

5.1. Détails du prototype et estimation du bruit

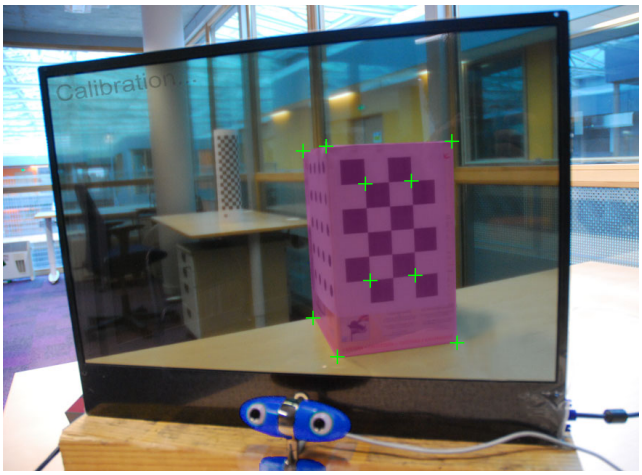


Figure 7. Prototype utilisé pour les expérimentations. Les points de référence utilisés sont indiqués par une croix. Le dispositif de localisation de l'utilisateur est ici une caméra stéréo (au premier plan)

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 3D de référence de l'environnement $\mathbf{\Sigma}_O$ et les observations 2D de l'utilisateur $\mathbf{\Sigma}_c$. Nous définissons ici la configuration du prototype (figure 7) utilisé pour l'évaluation.

POINTS 3D DE RÉFÉRENCE. Nous utilisons comme objet de référence une boîte rectangulaire de dimensions $372 \times 305 \times 229$ mm, placée à environ 1 mètre de l'écran. 10 points de référence sont choisis : les 6 coins visibles ainsi que 4 points additionnels sur les côtés. L'objet est statique (configuration *vitrine augmentée* section 4) et le repère $\mathcal{C}_o$ est défini comme les coordonnées locales de l'objet. Il n'y a donc pas de dispositif de localisation dans l'environnement et le bruit sur les points de référence est donc uniquement imputable aux erreurs du modèle 3D. En pratique ces dernières sont très faibles, on fixe expérimentalement :

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

POSITIONS DE L'UTILISATEUR. Il n'y a aucune contrainte sur la méthode de suivi utilisée pour l'utilisateur. Il est par exemple possible d'utiliser une solution commerciale monoculaire telle que `faceAPI`[4]. Nous utilisons dans nos expérimentations une caméra stéréoscopique USB Minoru (deux images de 640×480 pixels). La pupille gauche de l'utilisateur est détectée dans chaque caméra (Viola, Jones, 2001), puis un algorithme Mean-Shift (Comaniciu *et al.*, 2000) permet de trouver le centre de la

4. http://www.seeingmachines.com/product/faceapi/

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 très bruitée. Pour modéliser le bruit, nous avons observé le comportement de l'estimation pour des utilisateurs immobiles pendant plusieurs centaines d'images. La variance observée est :

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

Lors de la mise en œuvre de l'étalonnage, l'utilisateur doit théoriquement rester parfaitement immobile pendant chaque série de clics, ce qui est impossible. Pour limiter la perte de précision, sa position est mesurée à chaque clic et la moyenne est calculée pour chaque série. 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 :

$$\mathbf{\Sigma}_{U_{\text{average}}}^{(n=10)} = \mathrm{diag}(5^2/n, 5^2/n, 20^2/n) = \mathrm{diag}(1.6^2, 1.6^2, 6.3^2) \text{ en millimètres} \tag{20}$$

La caméra a un champ de vision restreint qui empêche le suivi de l'utilisateur hors de la zone définie par :

$$U_i^{(\mathcal{C}_u)} \in [-300, 300] \times [-150, 150] \times [400, 1000] \text{ en mm} \tag{21}$$

Cette zone est encore réduite par la contrainte que tous les points 3D de référence doivent être visibles à travers l'écran. Le nombre de points de vue différents pour l'évaluation est fixé à 20.

OBSERVATIONS 2D DE L'UTILISATEUR. L'écran transparent est un écran SAMSUNG 22 pouces de référence LTI220MT02. La surface active est de 473.6×296.1 mm^2 avec une résolution de 1680×1050 pixels. La taille d'un pixel est donc $s_{\text{px}} = 0.282$ mm/px.

L'utilisateur indique la position des observations 2D des points de référence dans l'écran par le biais de clics de souris. Notons qu'en pratique ces clics sont imprécis à cause de plusieurs facteurs : 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 sur plusieurs utilisateurs a montré que l'erreur moyenne est d'environ 3 pixels.

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

5.2. Évaluation sur données synthétiques

Pour étudier le comportement de l'initialisation, une scène synthétique est générée avec une géométrie et des niveaux de bruit similaires au cas réel présenté en section 5.1.

À partir de cette référence réaliste, l'influence des différents paramètres est mesurée sur 50 tirages aléatoires avant de calculer la moyenne et l'écart-type de l'erreur. L'erreur 3D de l'étalonnage en rotation et translation est utilisée plutôt que l'erreur 2D de reprojection dans les caméras virtuelles qui n'est pas un bon indicateur de la qualité de l'initialisation. Les lignes 1 et 2 de la figure 8 montrent que le bruit sur les positions utilisateur n'affecte pas les caméras centrées sur ces dernières, et le bruit sur les points 3D de référence n'affecte pas les caméras centrées sur les points de référence. Le résultat le plus intéressant de ces expériences est l'importance cruciale de la géométrie du problème (théoriquement justifié en section 3.6.1). En effet elle est mise en évidence par les courbes sur l'échelle de l'objet (ligne 3) : l'erreur en translation de la pose du dispositif de localisation dans l'environnement est par exemple triplée lorsque la taille de l'objet est divisée par deux.

La précision de l'ajustement de faisceaux est évaluée sur la figure 9. L'erreur de reprojection est presque constante à 3 pixels. Or il se trouve que 3 pixels est la valeur de l'écart-type du bruit sur les observations 2D, c'est donc l'erreur minimale atteignable. Après l'ajustement de faisceaux, la solution est bien optimale.

5.3. Évaluation sur données réelles

Le processus d'étalonnage est ensuite mis à l'épreuve sur le prototype réel visible sur la figure 7. Un utilisateur clique à la souris sur 10 points précédemment repérés sur la boîte et répète l'opération depuis 20 points de vue différents. Qualitativement, l'erreur d'alignement est presque imperceptible comme le montrent les exemples de reprojections de la figure 10. Des résultats quantitatifs sont produits au moyen d'une validation croisée : 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 sur la figure 11. L'erreur de reprojection moyenne est d'environ 10 pixels ce qui correspond à moins de 3 mm à l'écran.

5.4. Comparaison avec travaux similaires

Il est intéressant de comparer ce résultat d'étalonnage aux méthodes utilisant un miroir (Rodrigues *et al.*, 2010 ; Takahashi *et al.*, 2012). Ces deux publications présentent des résultats en conditions réelles comparables et rapportent respectivement une erreur de $7^\circ = 0.122\,\mathrm{rad}$ (Rodrigues *et al.*, 2010) et $3.26^\circ = 0.057\,\mathrm{rad}$ (Takahashi *et al.*, 2012).

La figure 12 montre que dans le cas d'une erreur angulaire α faible, pour un utilisateur à une distance d_U de l'écran et un point 3D à une distance d_O de l'écran, l'erreur de reprojection minimale résultante est $e = \frac{\alpha d_U d_O}{d_U + d_O}$. Dans le cas d'un utilisateur à $d_U = 0.7$ mètres de la caméra (position moyenne dans notre configuration) et d'un point de référence à $d_O = 1$ mètre de l'écran, une erreur angulaire de $0.06\,\mathrm{rad}$ se traduit par une erreur d'alignement de $\frac{0.06 \times 0.7 \times 1}{0.7+1} \approx 0.025$ mètres (25 mm). Cette borne inférieure de l'erreur, sans prendre en compte l'erreur en translation et les erreurs

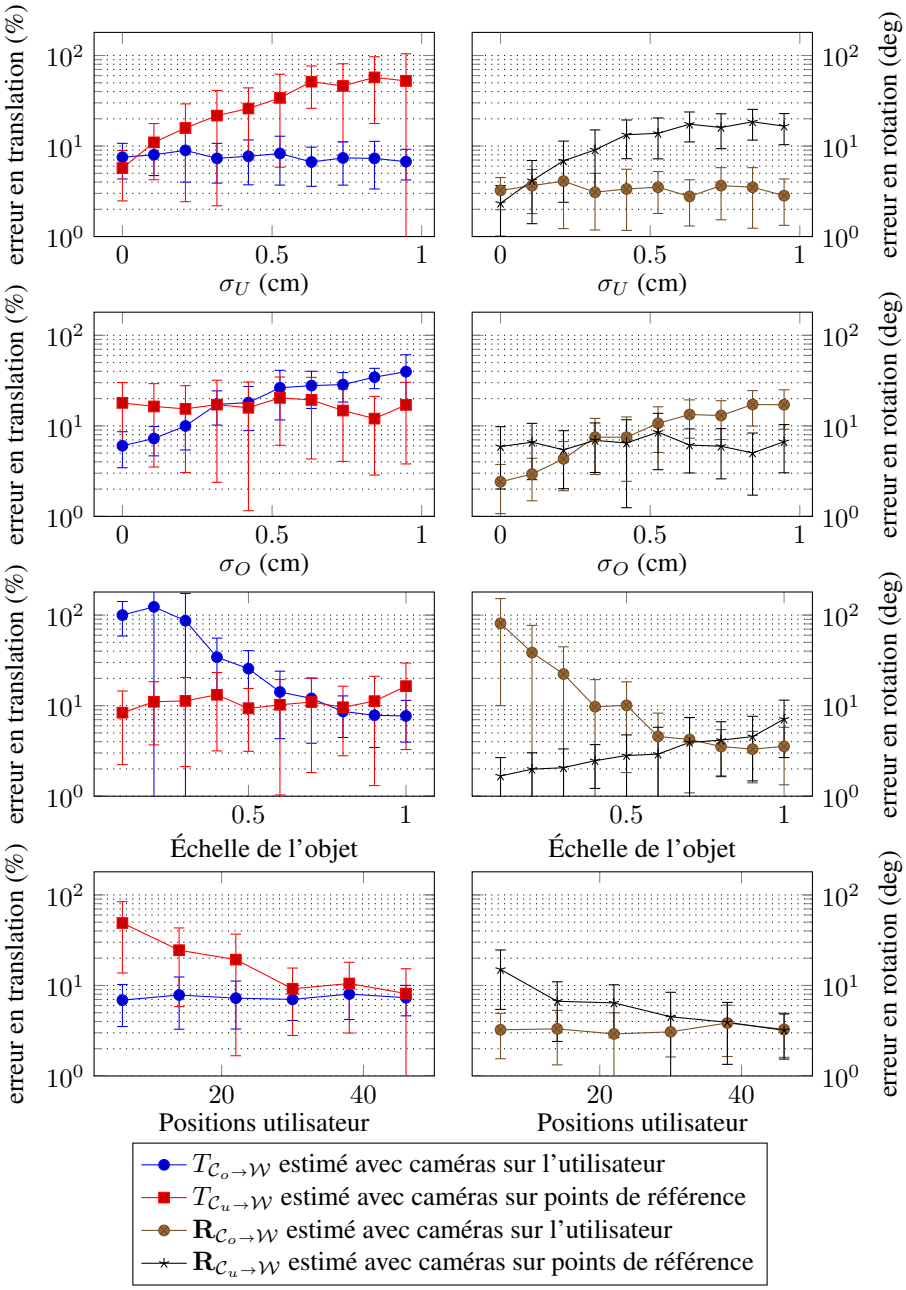


Figure 8. Comparaison de la robustesse au bruit sur les positions utilisateur, positions des points 3D de référence de l'environnement, échelle de l'objet et nombre de positions utilisateur. Les points correspondent à la valeur moyenne sur 50 échantillons et les barres correspondent à l'écart-type. L'échelle en y est logarithmique

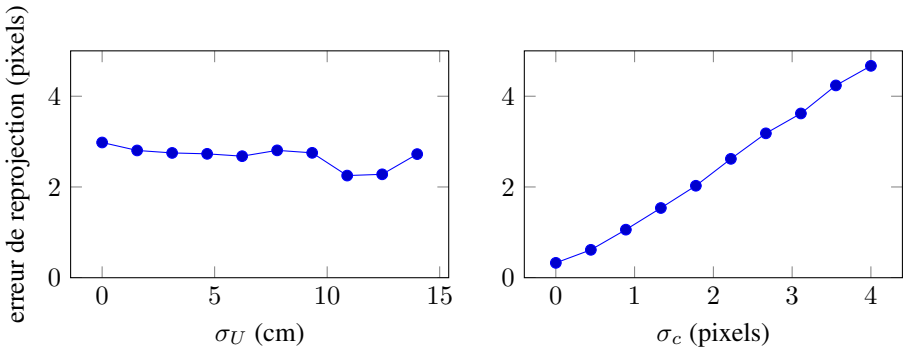


Figure 9. Ajustement de faisceaux sur données synthétiques

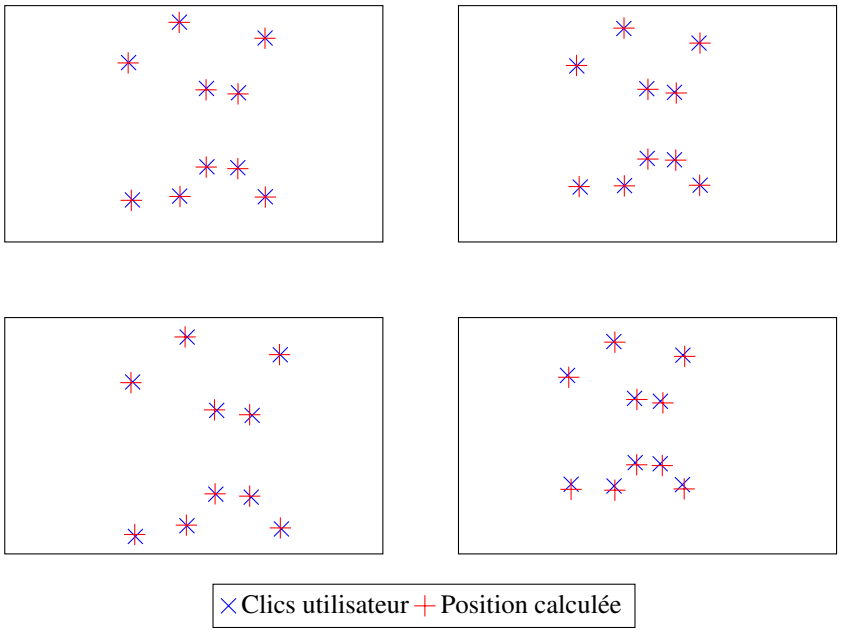


Figure 10. Exemples d'alignement sur les données de validation croisée

introduites par les méthodes de localisation, est plus de 7 fois supérieure à nos résultats expérimentaux (3 mm). Cette erreur est beaucoup trop importante pour des applications de réalité augmentée. De plus, comme précisé en introduction, ces méthodes ne sont applicables qu'à des caméras alors que notre méthode est indépendante des dispositifs de localisation utilisés.

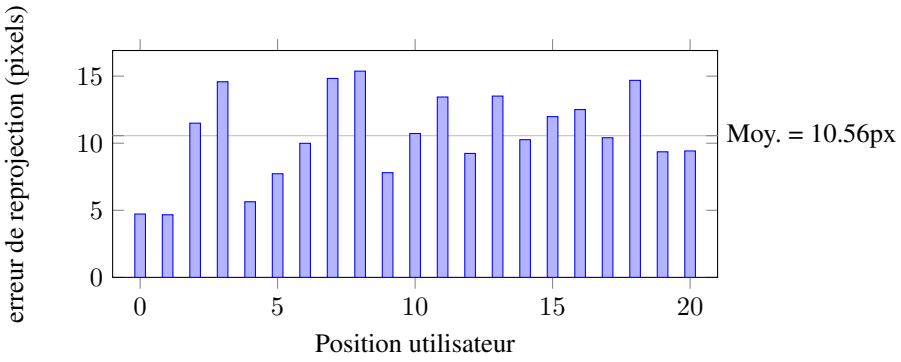


Figure 11. Validation croisée

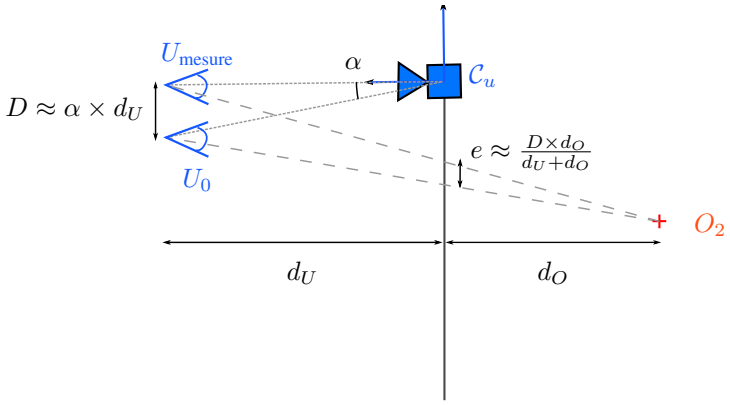


Figure 12. Étude de l'impact d'une erreur angulaire pour les méthodes estimant la pose du dispositif de localisation par rapport à l'écran. Ces relations sont uniquement valables pour de faibles valeurs de l'erreur angulaire α mais permettent d'estimer l'erreur d'alignement minimale e

6. Conclusion

Cet article présente une solution à l'étalonnage des systèmes de réalité augmentée utilisant un affichage semi-transparent et deux dispositifs de localisation (utilisateur et environnement). Notre première contribution consiste en la formalisation du problème de calibration d'un tel système, menant à la formulation d'un ajustement de faisceaux pour obtenir la solution optimale. Notre deuxième contribution consiste en l'introduction d'un nouveau formalisme autour de caméras virtuelles afin d'obtenir une initialisation convexe à cet ajustement de faisceaux. Les expériences ont démontré la précision et la robustesse de l'approche, ainsi que sa généralisation à d'autres configurations. Nous envisageons par la suite d'étudier différents types de solutions de localisation afin de mettre en place 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.

Remerciements

Ce travail a été financé en partie par la bourse de recherche ERC 307483 FLEXABLE du programme FP7 de l'union européenne.

Bibliographie

Abdel-Aziz Y., Karara H. (1971). Direct linear transformation from comparator to object space coordinates in close-range photogrammetry. In *ASP symposium on close-range photogrammetry*, p. 1–18. American Society of Photogrammetry.

Anjum N., Taj M., Cavallaro A. (2007). Relative position estimation of non-overlapping cameras. In *International Conference on Acoustics, Speech and Signal Processing*, vol. 2, p. II-281–II-284. IEEE Signal Processing Society.

Caspi Y., Irani M. (2002). Aligning non-overlapping sequences. *International Journal of Computer Vision*, vol. 48, n° 1, p. 39–51.

Comaniciu D., Ramesh V., Meer P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition*, vol. 2, p. 142–149. IEEE Computer Society.

Esquivel S., Woelk F., Koch R. (2007). Calibration of a multi-camera rig from non-overlapping views. In F. Hamprecht, C. Schnörr, B. Jähne (Eds.), *Pattern recognition*, vol. 4713, p. 82–91. Berlin / Heidelberg, Springer.

Hartley R. I., Zisserman A. (2004). *Multiple view geometry in computer vision* (Seconde éd.). Cambridge University Press.

Horn B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, vol. 4, n° 4, p. 629–642.

Juniper Research. (2012). *Over 2.5 billion mobile augmented reality apps to be installed per annum by 2017.* Press Release. Consulté sur http://www.juniperresearch.com/viewpressrelease.php?pr=334

Kumar R. K., Ilie A., Frahm J.-M., Pollefeys M. (2008). Simple calibration of non-overlapping cameras with a mirror. In *Computer Vision and Pattern Recognition*, p. 1–7. IEEE Computer Society.

Lébraly P., Royer E., Ait-Aider O., Deymier C., Dhome M. (2011). Fast calibration of embedded non-overlapping cameras. In *International Conference on Robotics and Automation*, p. 221–227. IEEE Robotics & Automation Society.

Marquardt D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, n° 2, p. 431–441.

Oliensis J., Hartley R. (2005). Iterative extensions of the Sturm/Triggs algorithm: Convergence and nonconvergence. *Pattern Analysis and Machine Intelligence*, vol. 29, n° 12, p. 2217–2233.

Rahimi A., Dunagan B., Darrell T. (2004). Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition.* IEEE Computer Society.

Rodrigues R., Barreto J., Nunes U. (2010). Camera pose estimation using images of planar mirror reflections. In *European Conference on Computer Vision*, p. 382–395. Springer.

Sturm P., Bonfort T. (2006). How to compute the pose of an object without a direct view? In *Asian Conference on Computer Vision*, vol. 2, p. 21–31. Springer.

Takahashi K., Nobuhara S., Matsuyama T. (2012). A new mirror-based extrinsic camera calibration using an orthogonality constraint. In *Computer Vision and Pattern Recognition.* IEEE Computer Society.

Tang A. (2003). Evaluation of calibration procedures for optical see-through head-mounted displays. In *2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, p. 161. IEEE Computer Society.

Triggs B. (1996). Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition*, p. 845–851. IEEE Computer Society.

Triggs B., McLauchlan P. F., Hartley R. I., Fitzgibbon A. W. (2000). Bundle adjustment — a modern synthesis. In *Vision algorithms: theory and practice*, p. 298–372. Springer.

Viola P., Jones M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, p. I-511–I-518. IEEE Computer Society.

Zhang Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence*, vol. 22, n° 11, p. 1330–1334.

Article reçu le 25/09/2013
Accepté le 12/05/2014

Annexe A. Caméras virtuelles centrées sur les points de référence

Il est possible d'effectuer l'étalonnage de caméras virtuelles (voir section 3.2) centrées sur les points 3D de référence de l'environnement. Le problème est symétrique avec quelques changements. Comme leurs axes optiques sont dans la direction $-z$, les poses des caméras virtuelles sont définies par $\mathbf{M}^{O}{}_{j}^{(\mathcal{C}_o)} = [\mathbf{R}^O | O_j^{(\mathcal{W})}]$ où $\mathbf{R}^O = \mathrm{diag}(-1, 1, -1)$ et $O_j^{(\mathcal{W})} = \mathbf{M}_{\mathcal{C}_o \to \mathcal{W}} O_j^{(\mathcal{C}_o)}$. L'équation (8) devient :

$$\mathbf{K}_j^O = \begin{bmatrix} \left[O_j^{(\mathcal{W})}\right]_z & 0 & -\left[O_j^{(\mathcal{W})}\right]_x \\ 0 & \left[O_j^{(\mathcal{W})}\right]_z & \left[O_j^{(\mathcal{W})}\right]_y \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

et les projections des points $U_{i=1\ldots m}^{(\mathcal{C}_u)}$ dans la caméra virtuelle j sont

$$c_{ij}^O = (-(c_{ij})_x, (c_{ij})_y)^T. \tag{24}$$

Le reste du processus de résolution est identique et permet d'obtenir une estimation $\widehat{\mathbf{M}}_{\mathcal{C}_u \to \mathcal{W}}$ de la pose de $\mathcal{C}_u$.