Plus de robustesse et de précision dans le suivi en temps réel de motifs plans

More robustness and precision in the real-time tracking of planar targets

Matthijs Douze[1] Bernard Thiesse[1] Vincent Charvillat[1]
[1] IRIT UMR CNRS 5505

ENSEEIHT, 2 rue Camichel, BP 7122, 31071 Toulouse Cedex 7
douze,charvi,thiesse@enseeiht.fr

Résumé

Nous présentons une étude comparative de trois algorithmes de suivi à base d'homographies qui peuvent s'exécuter en temps réel : un algorithme non linéaire (LM), la méthode de Hager et Belhumeur (HB) et celle de Jurie et Dhome (JD).

Après une analyse du contexte géométrique, nous aboutissons à un problème d'optimisation qu'il faut résoudre pour chaque image, séquentiellement. Nous montrons que les résultats des trois méthodes sont des compromis robustesse/précision. Nous proposons des améliorations aux méthodes numériques de résolution associées aux algorithmes (HB) et (JD) (sur la base des « modèles avec erreurs dans les variables » – E.I.V.). En enchaînant ces algorithmes améliorés, nous atteignons simultanément les deux qualités précitées.

Mots Clef

suivi, images multiples, vision 3D et géométrie, estimation de paramètres

Abstract

In this study, we present a comparison of three homography-based tracking algorithms that can run in real time : a non-linear algorithm (LM), the Hager and Belhumeur (HB) algorithm, and that of Jurie and Dhome (JD).

The analysis of the geometrical context leads to an optimization problem that must be solved sequentially for each image. We show that the results of the three methods are robustness/precision tradeoffs. We suggest improvements of the numerical methods associated with algorithms (HB) and (JD) (based on "Errors-In-Variables modeling" – E.I.V.). By chaining the improved algorithms we achieve both of the aforementioned qualities.

Keywords

tracking, multiple views, 3D vision and geometry, parameter estimation

Introduction

La mise en correspondance d'indices visuels communs à plusieurs images est une des tâches fondamentales des chercheurs en vision par calculateur. Dans le domaine de la vidéo, cela débouche sur un problème de suivi : il s'agit de déterminer où apparaît l'image d'un objet dans les diverses vues de la séquence.

Cette aptitude à suivre automatiquement un objet trouve une application dans beaucoup de domaines, comme par exemple :

– le contrôle de production, pour vérifier le déplacement d'objets dans une usine ;
– la réalité augmentée, pour insérer ou retirer virtuellement des objets dans une scène ;
– la compression, pour coder séparément le fond et les objets dans une vidéo ;
– etc.

Beaucoup de méthodes ont été imaginées et mises en œuvre pour effectuer le suivi : Cavallero et al. [5] présentent un état de l'art en la matière. Les familles d'approches sont les suivantes :

– suivi basé sur les régions : on utilise la couleur ou la texture pour segmenter les images de la séquence ; ensuite on met en correspondance les régions extraites ;
– suivi basé sur les contours ou un maillage : on cherche à identifier et à suivre les contours des objets mouvants (« contours actifs ») ; les arêtes d'un maillage peuvent soit épouser un contour, soit dans le cas d'un modèle déformable, isoler les régions selon leur mouvement ;
– suivi basé sur des formes : on identifie sur les images des formes très locales (typiquement des coins) faciles à mettre en correspondance ;
– suivi basé sur des modèles : on met en correspondance les images avec des modèles 2D ou 3D extraits d'une base de données.

Notre approche est du genre « suivi basé sur les régions, » mais nous n'utilisons effectivement qu'un nombre restreint de points.

Ce papier est organisé de la façon suivante :

– la section 1 (« Modélisation ») définit le contexte phy-

sique (donc les contraintes) qui permettent de poser le problème en termes de problème d'optimisation ;
– la section 2 (« Résolution ») fournit les détails des trois algorithmes existants que nous cherchons à améliorer et à combiner ;
– la section 3 (« Notre algorithme ») est notre proposition d'intégration des trois algorithmes précédents ;
– la section 4 (« Expérimentations ») est consacrée à la présentation des expériences (protocoles, résultats, analyse critique de ceux-ci) menées sur notre sujet d'étude.

1 Modélisation

1.1 Le suivi

Les données du problème consistent en une séquence d'images $(I_t)_{t\in\mathbb{N}}$ et une région d'intérêt $\mathcal{R}_0$ sur I_0. Cette région est contenue dans l'image de l'objet de la scène qu'on veut localiser dans les vues suivantes.

Hypothèse géométrique. On suppose qu'on peut décrire le mouvement de l'image de l'objet par une fonction d'un vecteur de p paramètres. C'est-à-dire que, pour tout instant t et pour tout point $\mathbf{m} \in \mathcal{R}_0$, il existe $\beta_t \in \mathbb{R}^p$ tel que le point $\mathbf{m}'$ de I_t qui représente le même point de la scène que $\mathbf{m}$ ait pour coordonnées $\mathbf{m}' = T(\beta_t, \mathbf{m})$, où T est une fonction modélisant la transformation paramétrée par β_t. $T(\beta_0, .)$ est l'identité.
Cette supposition exclut *a priori* les occultations, qu'elles soient dues à l'objet lui-même, aux autres objets ou au champ visuel de la caméra.

Hypothèse photométrique. On suppose aussi que l'image d'un point donné de la scène ne change pas de niveau de gris. On parle d'« hypothèse de conservation des niveaux de gris », dont il existe deux versions : la **faible** et la **forte**.
L'hypothèse faible ([22] eq 1, [24] eq 1, [25] eq 5, [2] eq 1, [3] eq 3, [11] p490, [1] p240, [4] eq 7) est à la base de l'équation fondamentale du flux optique ([14], p282), à savoir la conservation des niveaux de gris entre deux images successives. Mathématiquement,

$$\forall t \in \mathbb{N}, \forall \mathbf{m} \in \mathcal{R}_0, I_{t-1}(T(\beta_{t-1}, \mathbf{m})) \approx I_t(T(\beta_t, \mathbf{m}))$$

où $I(\mathbf{m})$ est le niveau de gris de $\mathbf{m}$ sur l'image I (la relation $\approx$ n'est pas transitive).
L'hypothèse forte ([12] eq 1, [15] p60, [18] eq6) est l'hypothèse faible étendue à deux images quelconques de la séquence et en conséquence :

$$\forall t \in \mathbb{N}, \forall \mathbf{m} \in \mathcal{R}_0, I_0(\mathbf{m}) = I_t(T(\beta_t, \mathbf{m})) \qquad (1)$$

Nous utilisons l'hypothèse forte. Le suivi consiste donc à estimer les valeurs de β_t pour que l'équation (1) soit aussi bien vérifiée que possible.
$\mathcal{R}_0$ est un ensemble infini de points connexes. Sur calculateur, nous devons manipuler un sous-ensemble fini de $\mathcal{R}_0$: $\widetilde{\mathcal{R}_0} = \{\mathbf{m}_1, ..., \mathbf{m}_n\}$ est l'ensemble des **points de référence**.

Le problème d'estimation. Tous les ingrédients du problème de l'estimation des paramètres $\beta \in \mathbb{R}^p$ d'un système qui prend en entrée la variable explicative $X \in \mathbb{R}^m$ et renvoie en sortie la variable dépendante $Y \in \mathbb{R}^q$, peuvent être regroupés sous la forme graphique synthétique suivante :

$$X \in \mathbb{R}^m \rightarrow \boxed{\begin{array}{c}\beta \in \mathbb{R}^p \\ m = 2; n; \\ p; q = 1\end{array}} \rightarrow Y = I_t(T(\beta, X)) \in \mathbb{R}^q$$

L'estimation indirecte des β repose sur n expériences similaires consistant (pour $i \in 1..n$) à « introduire » une valeur $\widetilde{X_i} = \mathbf{m}_i$ à l'entrée du système et à « mesurer » le niveau de gris $\widetilde{Y_i} = I_0(\mathbf{m}_i)$ (par utilisation de l'hypothèse forte).

Le critère. Résoudre un problème d'estimation de paramètres, c'est trouver[1] $\widehat{\beta_t}$ qui permet de « réduire l'écart » entre les sorties prédites par le modèle, $I_t(T(\beta, \mathbf{m}_i))$ et les résultats attendus $I_0(\mathbf{m}_i)$ pour tout i.
Réduire l'écart au sens de la « régression en distance verticale associée à la norme euclidienne » consiste à minimiser (moindres carrés non linéaires classiques)

$$F_t^{\mathrm{NLS}}(\beta) = \sum_{\mathbf{m}\in\widetilde{\mathcal{R}_0}} (I_t(T(\beta, \mathbf{m})) - I_0(\mathbf{m}))^2$$

donc

$$\widehat{\beta_t} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} F_t^{\mathrm{NLS}}(\beta) \qquad (2)$$

1.2 Le contexte

Dans notre cas, nous souhaitons faire un suivi à base d'homographies. Pour utiliser ce modèle, il faut en plus des hypothèses précisées auparavant, se placer dans une des conditions expérimentales suivantes :
– l'objet suivi est rigide et plan ;
– le centre optique de la caméra ainsi que la scène sont immobiles.
Dans ce cas, $T(\beta, .)$ est une homographie ([13] p194) et le vecteur de paramètres $\beta = [h_{11} \quad \cdots \quad h_{32}]^\top$ contient ses coefficients :

$$T\left(\beta, \begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+1} \\ \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+1} \end{bmatrix}$$

On peut disposer les coefficients h_{ij} dans une matrice, en complétant avec $h_{33} = 1$. Inversement, la version **normalisée** de la matrice $H = [h_{ij}]_{i,j=1..3}$ est $\frac{1}{h_{33}}H$. L'ensemble des homographies est un groupe :
– la loi interne est notée $\beta_1\beta_2$; $\beta_1\beta_2$ a pour matrice le produit normalisé des matrices de β_1 et β_2 ;
– l'inverse (notée β^{-1}) a pour matrice la matrice inverse normalisée ;
– l'élément neutre est $\beta_0 = [1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^\top$.

[1] Nous notons désormais avec un chapeau $\widehat{\cdot}$ les valeurs estimées.

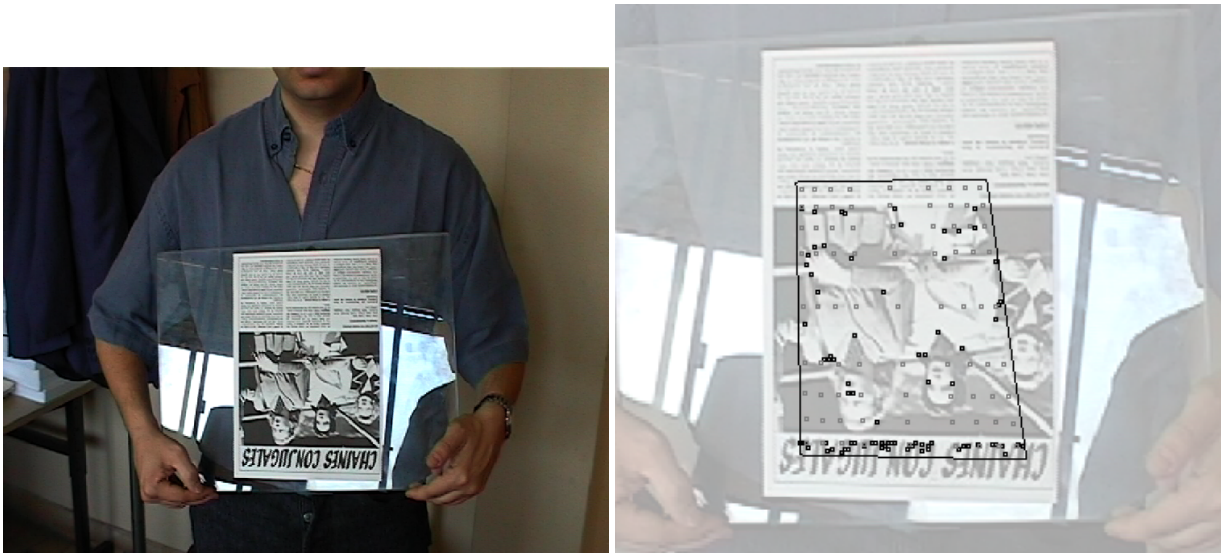

FIG. 1 – Une image d'une séquence vidéo et ses points de référence. Le cadre représente $\mathcal{R}_0$, les points d'intérêt sont en noir, les points réguliers en gris.

La région $\mathcal{R}_0$ est un quadrilatère dont les sommets sont définis par l'utilisateur ou une autre méthode *ad hoc*. Son image par $T(\beta_t, .)$ est aussi un quadrilatère.
On suppose par ailleurs qu'entre les prises de vue, les mouvements des objets et de la caméra sont **petits**. Ceci induit un écart $\beta_t - \beta_{t-1}$ faible.
Cette hypothèse est naturelle dans le domaine de la vidéo, où l'impression de mouvement est créée justement par les petites variations de la scène entre des images successives.

2 Résolution

Comme β_0 est connu et β_t peu différent de β_{t-1}, il est commode de faire un suivi « incrémental » : calculer $\widehat{\beta_t}$ revient à estimer $\Delta\beta_t = \beta_t - \beta_{t-1}$.
Nous supposons qu'il est possible de faire une **phase d'apprentissage**, arbitrairement longue, qui n'utilise que la première image. C'est à ce moment que les points de $\mathcal{R}_0$ retenus dans $\widetilde{\mathcal{R}_0}$ sont soigneusement choisis.
Ensuite, pour chaque nouvelle image, nous faisons les opérations de la **phase de suivi** qui permettent de calculer $\widehat{\Delta\beta_t}$, à l'aide des méthodes décrites dans la suite. Cette phase doit s'exécuter en temps réel.

2.1 Choix des points de référence

La tentation est grande de choisir les points d'intérêt habituellement utilisés dans la mise en correspondance d'images, parce qu'ils sont représentatifs de l'image. Cependant, comme ceux-ci s'accumulent autour des discontinuités de I_0, ils sont une source d'instabilité numérique. Nous utilisons donc un mélange de p_{h} points d'intérêt (de Harris, [23]) et de points répartis (dits **points réguliers**) arbitrairement sur $\mathcal{R}_0$. Ici, nous les choisissons répartis sur les nœuds d'une grille de taille $p_{\mathrm{r}} \times p_{\mathrm{r}}$ contenue dans $\mathcal{R}_0$ (figure 1).
Théoriquement, pour estimer les 8 paramètres de l'homographie, 4 points de référence suffisent (4 points $\rightarrow$ 8 coordonnées $\rightarrow$ 8 équations). Cependant, pour assurer la robustesse du suivi, nous en choisissons beaucoup plus, de l'ordre de quelques centaines.

2.2 Optimisation non linéaire

Le problème de minimisation de F_t^{NLS} est un problème de moindres carrés non linéaires et pour le résoudre, nous utilisons tout simplement l'algorithme de Levenberg-Marquardt (LM) ([8] eq 10.2.15). Notons r_t la fonction qui définit les résidus, à savoir :

$$r_t : \left\{ \begin{array}{ccc} \mathbb{R}^8 & \longrightarrow & \mathbb{R}^n \\ \beta & \longmapsto & \left[\begin{array}{c} I_t(T(\beta, \mathbf{m}_1)) - I_0(\mathbf{m}_0) \\ \vdots \\ I_t(T(\beta, \mathbf{m}_n)) - I_0(\mathbf{m}_n) \end{array} \right] \end{array} \right.$$

L'équation (2) devient alors :

$$\widehat{\beta_t} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \parallel r_t(\beta) \parallel_2^2$$

L'algorithme (LM) construit une suite $(\beta^i)_{i \in \mathbb{N}}$ à partir du point initial $\widehat{\beta_{t-1}}$ selon le schéma itératif suivant :

$$\left\{ \begin{array}{rcl} \beta^0 & = & \widehat{\beta_{t-1}} \\ \Delta^{\mathrm{LM}}\beta & = & (r_t'(\beta^i)^\top r_t'(\beta^i) + \alpha^i \mathrm{Id})^{-1} r_t'(\beta^i)^\top r_t(\beta^i) \\ \beta^{i+1} & = & \beta^i - \Delta^{\mathrm{LM}}\beta \end{array} \right.$$

où :
– $r_t'(\beta^i)$ est la matrice jacobienne de r_t évaluée en β^i ;
– $\alpha^i = 0$ si $\parallel \Delta^{\mathrm{GN}}\beta \parallel \leq \delta^i$, sinon α^i est l'unique valeur positive telle que $\parallel \Delta^{\mathrm{LM}}\beta \parallel = \delta^i$;

– le pas de Gauss-Newton est

$$\Delta^{\mathrm{GN}}\beta = r'_t(\beta^i)^+ r_t(\beta^i)$$

en notant $M^+ = (M^\top M)^{-1}M^\top$ la pseudo-inverse d'une matrice M de rang maximal ;
– δ^i est le rayon de la région de confiance à l'intérieur de laquelle l'approximation de r_t autour de β^i, par la partie régulière du développement de Taylor-Young à l'ordre 1, est acceptable.

Remarque 1 : Nous utilisons la mise en œvre de l'algorithme (LM) (gestion de δ^i et estimation de α^i) proposée par Moré et al. [19] et leur code MINPACK.

Remarque 2 : $\widehat{\beta_t}$ est la solution retournée par (LM) dont nous limitons les itérations à $n_{\mathrm{it}} \approx 10$.

Remarque 3 : La matrice jacobienne $r'_t(\beta)$ s'écrit :

$$r'_t(\beta) = \begin{bmatrix} I'_t(T(\beta,\mathbf{m}_1))^\top T_\beta(\beta,\mathbf{m}_1) \\ \vdots \\ I'_t(T(\beta,\mathbf{m}_n))^\top T_\beta(\beta,\mathbf{m}_n) \end{bmatrix} \quad (3)$$

où :
– $I'_t(\mathbf{m})$ est la dérivée spatiale de I_t évaluée en $\mathbf{m}$ (identifiable à un vecteur de $\mathbb{R}^2$) ;
– $T_\beta(\beta,\mathbf{m})$ est la dérivée partielle de T par rapport à β, évaluée en $(\beta,\mathbf{m})$ (identifiable à une martrice 2×8).

Le calcul de T_β ne pose pas de problème. Par contre, à cause de la nature discrète des images, il est nécessaire de travailler sur des valeurs interpolées de I_t (Cf. annexe).

2.3 Méthode de Hager et Belhumeur

Réécrivons l'équation (13) de l'article de Hager et Belhumeur [12] en utilisant nos notations, nous obtenons :

$$\Delta^{\mathrm{HB}}\beta = r'_{t-1}(\widehat{\beta_{t-1}})^+ r_t(\widehat{\beta_{t-1}})$$

$\Delta^{\mathrm{HB}}\beta$ serait le pas de la première itération de Gauss-Newton si r'_{t-1} était remplacée par r'_t.

Remarque 1 : Cette approximation faisant intervenir l'image I_{t-1} dans le calcul de la matrice jacobienne provient d'un développement de Taylor-Young à l'ordre 1 de I_t considérée dépendante de t et β.

Remarque 2 : Hager et Belhumeur proposent de pousser l'approximation plus loin en utilisant l'hypothèse forte en $t-1$. Ils aboutissent logiquement à une « estimation » de $r'_{t-1}(\widehat{\beta_{t-1}})$ indépendante de I_{t-1}, ne reposant que sur la dérivée de I_0, qui peut être précalculée pendant la phase d'apprentissage.

Avec nos notations, l'équation (19) de Hager et Belhumeur s'écrit :

$$r'_{t-1}(\widehat{\beta_{t-1}}) \approx \begin{bmatrix} I'_0(\mathbf{m}_1)^\top T_m(\widehat{\beta_{t-1}},\mathbf{m}_1)^{-1} T_\beta(\widehat{\beta_{t-1}},\mathbf{m}_1) \\ \vdots \\ I'_0(\mathbf{m}_n)^\top T_m(\widehat{\beta_{t-1}},\mathbf{m}_n)^{-1} T_\beta(\widehat{\beta_{t-1}},\mathbf{m}_n) \end{bmatrix}$$

où $T_m(\beta,\mathbf{m})$ est la dérivée partielle de T par rapport à $\mathbf{m}$ évaluée en $(\beta,\mathbf{m})$ (identifiable à une matrice 2×2).

2.4 La méthode de Jurie et Dhome

La méthode de Jurie et Dhome [15] commence par une coûteuse phase d'apprentissage reposant uniquement sur I_0.

Phase d'apprentissage. Le but de cette phase est d'éviter tout calcul de $r'_t(\beta)$ pendant le suivi, à la manière des variantes de Newton, dites « méthodes de Newton généralisées » ([6] p161) : c'est donc une extension de la proposition de Hager et Belhumeur.

Soit $\Delta\beta$ une « petite » perturbation de β_0. La partie régulière du développement de Taylor-Young à l'ordre 1 de r_0 autour de β_0 donne :

$$r_0(\beta_0+\Delta\beta) = r'_0(\beta_0)\Delta\beta \quad (4)$$

Étant donnée une valeur mesurée $\widetilde{r_{\Delta\beta}}$ de $r_0(\beta_0+\Delta\beta)$, on peut retrouver une approximation de $\Delta\beta$ par le pas de Gauss-Newton :

$$\Delta\beta \approx \Delta^{\mathrm{GN}}\beta = r'_0(\beta_0)^+ \widetilde{r_{\Delta\beta}}$$

Ceci montre qu'**une relation linéaire de matrice $A = r'_0(\beta_0)^+$ permet d'approcher la valeur de la perturbation à partir de la variation de niveaux de gris qu'elle induit**. C'est cette relation que Jurie et Dhome utilisent pendant la phase de suivi.

Cependant, plutôt que de calculer A selon l'expression suggérée ci-dessus (qui souffre du grand nombre d'approximations utilisées), ces auteurs proposent un processus d'estimation correspondant à la « boîte noire » suivante :

$$\Delta\beta \in \mathbb{R}^p \rightarrow \boxed{\begin{array}{c} A \in \mathbb{R}^{p\times n} \\ \hline m'=p; n'=N; \\ p'=p\times n; q'=p \end{array}} \rightarrow A\widetilde{r_{\Delta\beta}} - \Delta\beta = 0_{\mathbb{R}^p} \quad (5)$$

L'estimation de A repose sur N « expériences » consistant à tirer aléatoirement une perturbation $\widetilde{\Delta\beta_k}$ et à mesurer $r_0(\beta_0+\widetilde{\Delta\beta_k})$ noté $\widetilde{r_k}$.

Jurie et Dhome proposent une résolution à partir du problème aux moindres carrés suivant :

$$(P^{\mathrm{OLS}}) \left\{ \begin{array}{ll} \mathrm{Min} & \sum_{k=1}^n \parallel A\widetilde{r_k} - \widetilde{\Delta\beta_k} \parallel_2^2 \\ A \in \mathbb{R}^{p\times n} & \end{array} \right.$$

En notant :

$$B = \left[\widetilde{\Delta\beta_1} \quad \cdots \quad \widetilde{\Delta\beta_N}\right] = \begin{bmatrix} b_1 \\ \vdots \\ b_p \end{bmatrix}$$

$$\widetilde{R} = [\widetilde{r_1} \quad \cdots \quad \widetilde{r_N}]$$

$$A = \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}$$

on montre [10] que $\widehat{A^{\mathrm{OLS}}} = B\widetilde{R}^+$ déduite de :

$$\widehat{a_k^{\mathrm{OLS}}} = \left(\underset{a\in\mathbb{R}^n}{\mathrm{argmin}} \parallel \widetilde{R}^\top a - b_k^\top \parallel_2^2\right)^\top = \left(\underset{a\in\mathbb{R}^n}{\mathrm{argmin}} F_k^{\mathrm{LS}}(a)\right)^\top$$

$\forall k = 1..p$

Cette solution conduit à deux remarques :
– la matrice $\widetilde{R}^\top$ est commune à toutes les fonctionnelles aux moindres carrés F_k^{LS}. Seul le « second membre » $b_k^\top$ change !
– les mesures faites sur les « second membres » $b_k^\top$ sont sans erreur : ce sont de « bonnes valeurs aléatoires » ! Par contre, l'élément (i,j) de la matrice $\widetilde{R}$ fait intervenir le niveau de gris du point $T(\beta_0 + \widetilde{\Delta\beta_j}, \mathbf{m}_i)$: il est donc sujet à une imprécision liée à l'interpolation sous-jacente.

La conséquence logique de cette dernière remarque serait d'envisager la détermination de $\widehat{A}$ comme solution d'un problème (DLS) (« Data Least Squares ») ([7] eq4), pour lequel seul le second membre b' du système générique $(A'x' \approx b')$ est entaché d'erreurs.

Partant de là, la première remarque conduirait à se placer dans le cadre multidimensionnel (MDLS) et à envisager un problème de la forme $A'_{N\times n}X'_{n\times 8} \approx B'_{N\times 8}$ ([26] p50-51).

Dans les faits, les nombreuses expériences menées dans ce sens sur nos données, en collaboration avec Sabine Van Huffel et Yvan Markovsky (de l'université catholique de Louvain), nous permettent de conclure que le meilleur compromis, pour tenir compte de **la relative imprécision du modèle mathématique linéaire sous-jacent à l'équation (5)**, est de rechercher $\widehat{A}$ comme solution d'un problème de moindres carrés totaux pondérés dans le cas multidimensionnel (MWTLS) ([21] p14) et ([20] eq5, eq6).

La pondération est ciblée pour accorder une importance plus ou moins grande aux erreurs sur b' par rapport à celles sur A'. (P^{OLS}) est donc remplacé ([16], eq1) par

$$(P^{\mathrm{MWTLS}}) \begin{cases} \mathrm{Min} \parallel [\widetilde{\Delta R}^\top | \gamma \Delta B^\top] \parallel_{\mathrm{F}}^2 = F^{\mathrm{MWTLS}}(A) \\ A \in \mathbb{R}^{p\times N} \\ \widetilde{\Delta R} \in \mathbb{R}^{n\times N} \\ \Delta B \in \mathbb{R}^{p\times N} \\ (\widetilde{R} + \widetilde{\Delta R})^\top A^\top = (B + \Delta B)^\top \end{cases}$$

où :
– la norme matricielle $\parallel . \parallel_{\mathrm{F}}$ est celle de Frobénius ;
– $\gamma > 0$ est la pondération permettant de naviguer entre (LS) ($\gamma \to 0$) et (DLS) ($\gamma \to \infty$) en passant par (TLS) ($\gamma = 1$) ([20], p26).

En pratique :
1. $\gamma \in [10^{-4}, 10^{-2}]$ convient en général ;
2. la mise en œuvre est une adaptation (pour intégrer γ) de l'algorithme 3.1 de Van Huffel et Vanderwalle ([26] p87) qui transcrit le théorème 3.1 ([26] p52) dans le cas d'une solution unique.

Phase de suivi. On suppose que l'ensemble des paramètres est un groupe (on note l'opération comme un produit) dont la structure est préservée par T :

$$T(\beta_1, T(\beta_2, \mathbf{m})) = T(\beta_1\beta_2, \mathbf{m})$$

Nous avons vu (Cf. §1.2) que c'était le cas pour l'homographie.

Pendant la phase de suivi, à l'étape t, on mesure $r_t(\widehat{\beta_{t-1}})$, et on calcule $\beta'_0 = \beta_0 + \widehat{A} r_t(\widehat{\beta_{t-1}})$.

On a donc, pour $\mathbf{m} \in \widetilde{\mathcal{R}_0}$ (et par extension, pour $\mathbf{m} \in \mathcal{R}_0$),

$$I_t(T(\widehat{\beta_{t-1}}, \mathbf{m})) \approx I_0(T(\beta'_0, \mathbf{m}))$$

Posons $\beta = \widehat{\beta_{t-1}}\beta_0'^{-1}$ et vérifions, pour $\mathbf{m} \in \widetilde{\mathcal{R}_0}$:

$$\begin{aligned} I_t(T(\beta, \mathbf{m})) &= I_t(T(\widehat{\beta_{t-1}}\beta_0'^{-1}, \mathbf{m})) \\ &= I_t(T(\widehat{\beta_{t-1}}, T(\beta_0'^{-1}, \mathbf{m}))) \\ &\approx I_0(T(\beta'_0, h(\beta_0'^{-1}, \mathbf{m}))) \\ &\approx I_0(T(\beta'_0\beta_0'^{-1}, \mathbf{m})) \\ &\approx I_0(\mathbf{m}) \end{aligned}$$

donc β est une bonne approximation de β_t, si les mouvements ne sont pas trop grands :

$$\widehat{\beta_t} = \beta = \widehat{\beta_{t-1}}(\beta_0 + \widehat{A} r_t(\widehat{\beta_{t-1}}))^{-1}$$

Dans cette méthode, les paramètres susceptibles d'ajustement sont :
– le nombre d'expériences N faites pendant la phase d'apprentissage ;
– l'écart-type σ des perturbations appliquées à β_0. En pratique, nous ajoutons un bruit gaussien, centré d'écart-type σ, aux coordonnées des sommets du quadrilatère suivi ; β' (unique) correspond alors à l'homographie qui aurait induit cette perturbation et $\Delta\beta'_k = \beta' - \beta_0$.

2.5 Discussion

La méthode non linéaire :
– ne s'appuie sur aucune phase d'apprentissage ;
– nécessite une approximation initiale dans le domaine d'attraction de la solution comme tout algorithme apparenté à la méthode de Newton ;
– a un coût proportionnel au nombre d'itérations (le coût d'une itération est en $\mathcal{O}(np^2)$) ;
– est très précise.

La méthode de Hager et Belhumeur + WTLS a les caractéristiques suivantes :
– seule I'_0 peut être précalculée pendant la phase d'apprentissage ;
– la phase de suivi est relativement coûteuse à cause du calcul de la SVD ;
– une bonne approximation de la solution est d'autant plus nécessaire qu'il n'y a qu'une itération ;
– le résultat est précis.

À l'opposé, la méthode de Jurie et Dhome :
– est plus rapide lors de la phase de suivi ;
– requiert une phase d'apprentissage coûteuse ;
– dépend d'un paramètre σ qui permet d'ajuster le compromis robustesse/précision, en restant tributaire de l'approximation de modélisation, inhérente à l'équation (5). En augmentant σ, le suivi gagne en robustesse mais perd en précision et vice-versa.

Ces remarques, que nous avons déduites des algorithmes, sont confirmées par l'expérience (Cf. §4).

3 Notre algorithme

La méthode que nous proposons consiste à « enchaîner » les 3 algorithmes précédents pour passer de $\widehat{\beta_{t-1}}$ à $\widehat{\beta}_t$. Naturellement, nous commençons pas appliquer les algorithmes les plus robustes, avant d'utiliser les plus précis. L'algorithme est décrit dans les figures 2 et 3.

Phase d'apprentissage
entrées : $(\sigma_1, ..., \sigma_e), I_0, p_\mathrm{h}, p_\mathrm{r}$
avec $\sigma_1 > \sigma_2 > \cdots > \sigma_e$

$\widetilde{\mathcal{R}_0} \longleftarrow$ `points_choisis`$(I_0, p_\mathrm{h}, p_\mathrm{r})$
-- G contient $[I_0(\mathbf{m}_1) \ \cdots \ I_0(\mathbf{m}_n)]^\top$
$G \longleftarrow$ `échantillon`$(I_0, \widetilde{\mathcal{R}_0}, \beta_0)$
Pour i **de** 1 **à** e **faire**
-- phase d'apprentissage de Jurie et Dhome
-- écart-type σ_i sur les perturbations
$A_i \longleftarrow$ `apprentissage_jd`$(I_0, \sigma_i, \widetilde{\mathcal{R}_0})$
Finpour

FIG. 2 – Phase d'apprentissage de notre algorithme.

Phase de suivi
entrées : β_{t-1}, I_t.

-- ε correspond au β courant
$\beta \longleftarrow \beta_{t-1}$; $\varepsilon \longleftarrow F_t^{\mathrm{NLS}}(\beta)$
$\widetilde{r} \longleftarrow$ `échantillon`$(I_t, \widetilde{\mathcal{R}_0}, \beta) - G$
-- e itérations de Jurie et Dhome
Pour i **de** 1 **à** e **faire**
$\beta' \longleftarrow \beta(\beta_0 + A_i\widetilde{r})^{-1}$
$\varepsilon' \longleftarrow F_t^{\mathrm{NLS}}(\beta')$
Si $\varepsilon' < \varepsilon$ **alors**
$\beta \longleftarrow \beta'$; $\varepsilon \longleftarrow \varepsilon'$
$\widetilde{r} \longleftarrow$ `échantillon`$(I_t, \widetilde{\mathcal{R}_0}, \beta) - G$
Finsi
Finpour
-- une itération de Hager et Belhumeur
$\beta' \longleftarrow$ `suivi_hb`$(I_t, \widetilde{\mathcal{R}_0}, \beta)$
$\varepsilon' \longleftarrow F_t^{\mathrm{NLS}}(\beta')$
Si $\varepsilon' < \varepsilon$ **alors**
$\beta \longleftarrow \beta'$; $\varepsilon \longleftarrow \varepsilon'$
$\widetilde{r} \longleftarrow$ `échantillon`$(I_t, \widetilde{\mathcal{R}_0}, \beta) - G$
Finsi
-- quelques itérations non linéaires
$\beta' \longleftarrow$ `suivi_nl`$(I_t, \widetilde{\mathcal{R}_0}, \beta)$
$\varepsilon' \longleftarrow F_t^{\mathrm{NLS}}(\beta')$
Si $\varepsilon' < \varepsilon$ **alors**
$\beta \longleftarrow \beta'$; $\varepsilon \longleftarrow \varepsilon'$
$\widetilde{r} \longleftarrow$ `échantillon`$(I_t, \widetilde{\mathcal{R}_0}, \beta) - G$
Finsi
$\beta_t \longleftarrow \beta$

FIG. 3 – Phase de suivi de notre algorithme

Les fonctions qui apparaissent dans le code sont :
- `points_choisis`$(I, p_\mathrm{h}, p_\mathrm{r})$: renvoie les points de référence (Cf. §2.1) ;
- `échantillon`$(I, \widetilde{\mathcal{R}_0}, \beta)$: mesure les niveaux de gris de l'image I aux points $(T(\beta, \mathbf{m}))_{\mathbf{m}\in\widetilde{\mathcal{R}_0}}$. Nous utilisons la « méthode brute » (annexe 4.2) pour évaluer I ;
- `apprentissage_jd`(I, σ, K) : effectue la phase d'apprentissage de Jurie et Dhome **améliorée par la résolution en MWTLS** (Cf. §2.4), avec un écart-type σ pour les perturbations. Le nombre d'expériences est $N = 5000$;
- `suivi_hb`$(I, \widetilde{\mathcal{R}_0}, \beta)$: l'étape de suivi de Hager et Belhumeur **améliorée par la résolution en WTLS** (Cf. §2.3), en utilisant une convolution ($\sigma = 1, k = 1, d = 1$) pour dériver I ;
- `suivi_nl`$(I, \widetilde{\mathcal{R}_0}, \beta)$: entre 2 et 20 itérations d'estimation (Cf. §2.2). I_t et I'_t sont évaluées par interpolation bilinéaire.

4 Expérimentation

Nous avons fait des expériences d'abord sur des images de synthèse pour optimiser les paramètres des algorithmes, puis sur des images réelles.
La mise en œuvre de notre algorithme en langage C utilise les bibliothèques en langage Fortran suivantes :
- MINPACK [19] pour la mise en œuvre de Levenberg-Marquardt ;
- LAPACK pour les décompositions QR ;
- VANHUFFEL pour les résolutions TLS.

Les trois sont disponibles sur Netlib (`http://netlib.enseeiht.fr`).

4.1 Images de synthèse

Nous avons choisi une image *a priori* idéale pour faire le suivi (figure 4). Nous avons ensuite perturbé l'image de trois manières différentes :
- Rotation : de 1° à 20° entre deux images successives.
- Zoom : nous alternons entre agrandissement et rapetissement de l'image, jusqu'à un facteur 2.5 et par pas variant de 1% à 20%.
- Déformation : le motif est déformé comme si son antécédent 3D avait subi une rotation autour d'un axe coplanaire d'angle variant entre -40° et +40°.

Les perturbations sont de plus en plus intenses au fur et à mesure du suivi. Par exemple, pour la séquence rotation, l'angle de rotation θ entre deux images est (en degrés) : $\theta = \frac{t}{20} + 1$
Comme nous disposons ici de la « vérité terrain, » nous pouvons mesurer directement l'erreur de suivi. Si β_t est la tranformation réelle, $\widehat{\beta}_t$ la version estimée et $(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4)$ les sommets du quadrilatère suivi, l'erreur (critère géométrique) est définie par :

$$\varepsilon_t = \sum_{i=1}^{4} \| \mathbf{w}_i - T^{-1}(\beta_t, T(\widehat{\beta}_t, \mathbf{w}_i)) \|^2$$

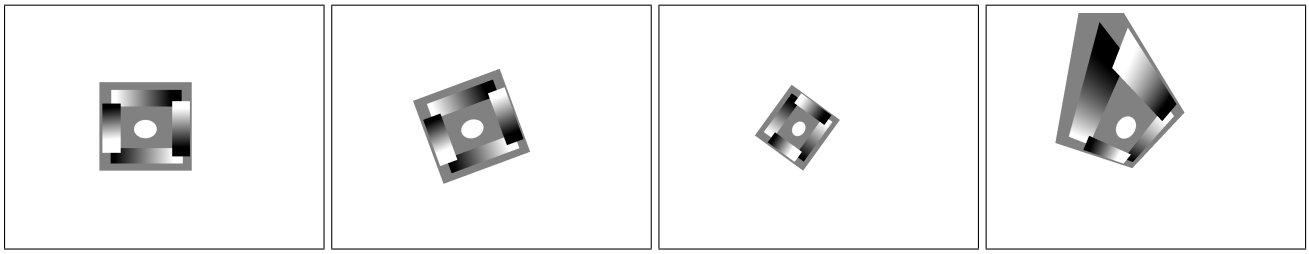

FIG. 4 – FIG : De gauche à droite : l'image synthétique utilisée pour les tests, l'image après rotation, zoom et déformation. L'image fait 1024×768 pixels. La région d'intérêt $\mathcal{R}_0$ est choisie à l'intérieur du motif central texturé.

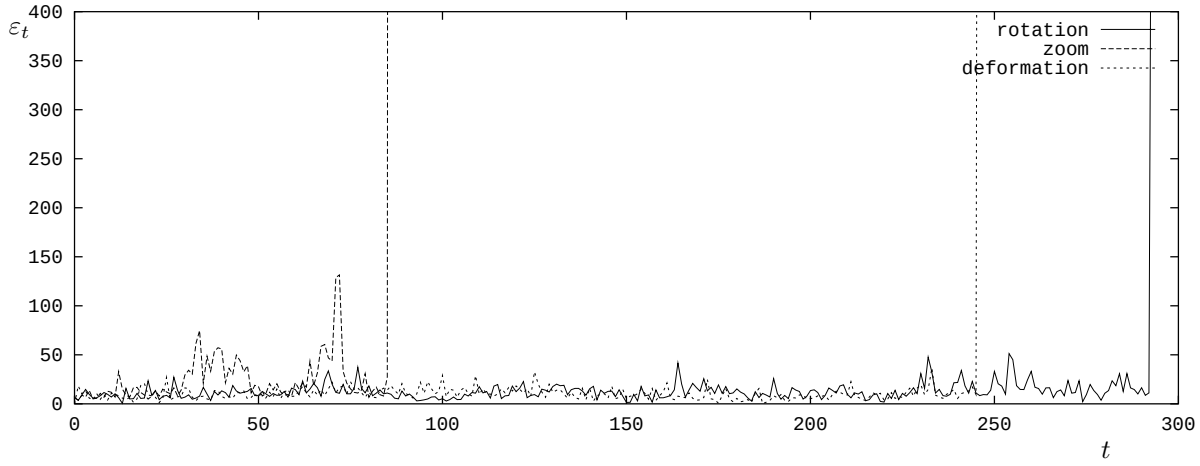

FIG. 5 – Évolution de l'erreur ε_t en fonction du temps pour les trois séquences, avec notre algorithme.

Un seuil s fixe l'erreur au-delà de laquelle l'objet est « perdu ». Comme les mouvements sont de plus en plus intenses, le suivi est de plus en plus difficile. La qualité du suivi est caractérisée par le t^* minimal satisfaisant $\varepsilon_{t^*} > s$. Nous choisissons $s = 400 = 4 \times 10^2$, ce qui représente une erreur quadratique moyenne de 10 pixels par sommet.

Pour notre algorithme, les résultats sont présentés à la figure 5. On voit que le suivi est toujours très précis jusqu'au moment où il échoue.

Influence du nombre de points de référence.

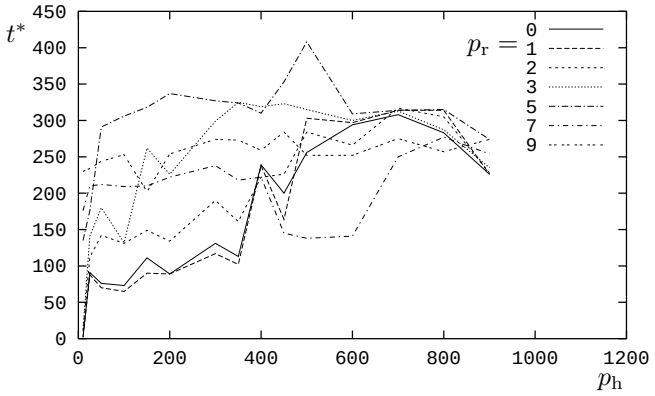

FIG. 6 – Durée t^* du suivi réussi en fonction du nombre de points d'intérêt p_h et de points réguliers p_r^2.

Nous avons cherché à quantifier le nombre de points de référence nécessaires pour le suivi le plus robuste. Pour cela, nous utilisons **notre algorithme** sur la séquence « rotation », en faisant varier le nombre p_r^2 de points réguliers et le nombre p_h de points d'intérêt. Le résultat est décrit dans la figure 6.

Les résultats sont assez instables. Cependant, on peut voir que :

– pour un nombre de points réguliers donné, il y a une valeur optimale du nombre de points d'intérêt ;
– la valeur optimale de p_r est 5.

Pour cette séquence, les valeurs optimales sont donc $(p_\mathrm{h}, p_\mathrm{r}) = (500, 5)$. Les résultats sont semblables pour les deux autres séquences.

Influence du choix des algorithmes. Nous avons fait des tests en n'appliquant qu'un algorithme par image de la séquence « déformation ». On voit sur la figure 7 que la robustesse de l'algorithme est caractérisée par le nombre (t^*) d'itérations pendant lesquelles ε_t ne dépasse pas le seuil. Sa précision est donnée par la valeur moyenne $\bar{\varepsilon}$ de ε_t, calculée entre $t = 0$ et $t = t^* - 1$.

Nous avons ensuite « enchaîné » les algorithmes de diverses manières, pour observer l'effet sur t^* et $\bar{\varepsilon}$. Les résultats sont résumés dans le tableau 1.

On voit que l'enchaînement qui offre le meilleur compromis robustesse/précision correspond à notre algorithme

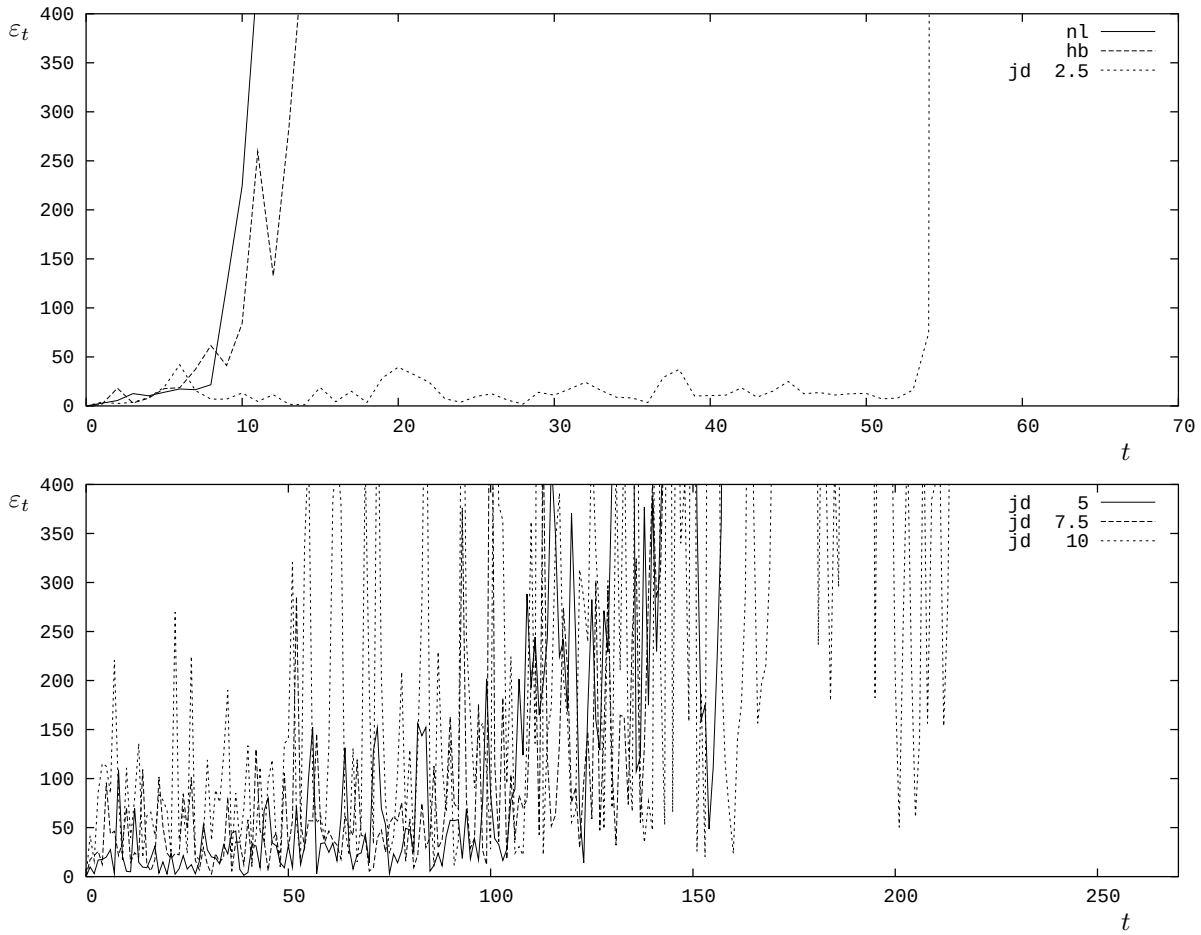

FIG. 7 – Évolution de l'erreur au cours du temps en utilisant un seul algorithme. `nl`=non linéaire, $n_{\text{it}} = 4$, `hb`=Hager et Belhumeur, `jd` σ = Jurie et Dhome avec une perturbation σ pendant l'apprentissage.

avec :

$$e = 4, (\sigma_1, \sigma_2, \sigma_3, \sigma_4) = (15, 10, 5, 2.5)$$

séquence	t^*	$\bar{\varepsilon}$
nl	11	40.66
hb	11	70.46
jd 5	85	98.41
hb, nl	19	49.17
jd 5, nl	111	106.84
jd 5, hb	111	30.82
jd 5, hb, nl	121	27.04
jd 5, jd 2.5, hb, nl	134	14.34
jd 10, jd 5, jd 2.5, hb, nl	149	14.98
jd 12.5, jd 10, jd 5, jd 2.5, hb, nl	238	28.10
jd 15, jd 10, jd 5, jd 2.5, hb, nl	278	31.67
jd 20, jd 10, jd 5, jd 2.5, hb, nl	112	10.09

TAB. 1 – Durée t^* (en images) du suivi réussi et erreur moyenne (en pixels) $\bar{\varepsilon}$ pour différentes combinaisons de méthodes (en utilisant la notation compacte de la figure 7).

Apport de WMTLS. Nous avons comparé les performances de notre algorithme en utilisant la méthode OLS ou WMTLS pendant la phase d'apprentissage des étapes de Jurie et Dhome. Ces performances dépendent fortement du nombre d'expériences (figure 8). La méthode WMTLS est en général plus efficace, même si ses performances s'estompent pour les valeurs élevées de N.

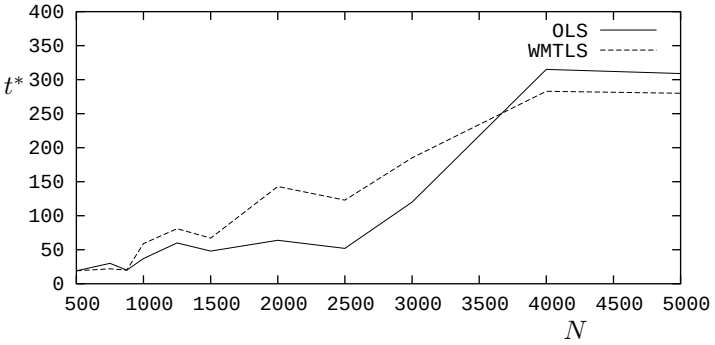

FIG. 8 – Durée t^* du suivi réussi selon la méthode d'estimation de la matrice A

nombre de points de référence	125	175	225	325	425	525
`échantillon`	0.074	0.113	0.127	0.181	0.238	0.289
calcul de F_t^{NLS}	0.012	0.015	0.018	0.027	0.031	0.038
calcul de Jurie et Dhome	0.209	0.237	0.264	0.351	0.415	0.490
`suivi_bh`	1.269	1.741	2.272	3.269	4.317	5.328
`suivi_nl`	1.527	2.024	2.640	3.781	5.095	6.020

TAB. 2 – Temps d'exécution pour les diverses étapes du suivi (figure 2), en millisecondes. Nous les avons mesurés sur notre combinaison d'algorithmes, en fonction du nombre de points de référence.

FIG. 9 – Images de séquences vidéo prises par une caméra DV, suivies par notre algorithme.

4.2 Images réelles

Mise en œuvre. Nous traitons en temps réel un flux vidéo DV sur un PowerPC G4 866, en visualisant les résultats à l'aide d'une petite interface. Nous avons accéléré la phase d'apprentissage en tirant parti des instructions vectorielles du processeur. Elle prend donc de 3 à 10 secondes selon le nombre de points de référence.

Les images arrivent à une fréquence de 25 Hz, mais leur traitement doit durer moins de 20 ms parce que la décompression DV et l'affichage prennent les 20 ms restantes. Nous ne pouvons pas nous permettre de sauter des images, parce que des mouvements brutaux de la scène pourraient nous échapper.

Comme on le voit dans le tableau 2, les opérations sont assez rapides pour faire plusieurs étapes d'estimation pour chaque image. La méthode de Jurie et Dhome est la plus rapide. La méthode de Hager et Belhumeur est plus lente, parce qu'elle nécessite la décomposition SVD d'une matrice $n \times (p+1)$. La méthode non linéaire, bien que plus lente, demeure dans les délais. Une séquence vidéo d'exemple est disponible à [27].

Les différents types de mouvement sont présentés (translation, rotation, zoom, etc.) Le suivi est correct pendant les 360 images de la séquence.

Exemples. Nous avons fait les expériences en déplaçant devant la caméra une surface plane, plus ou moins vigoureusement (figure 9), tout en effectuant des zooms.

En tournant les actes de RFIA 2002 (première image) en tous sens, nous nous sommes rendus compte de l'importance de la planéité de la surface. Nous avons donc collé une feuille texturée et mate sur un carreau de verre. Le suivi devient alors très fiable.

Par contre, la méthode n'est pas du tout robuste aux occultations : « presque tous » les points de référence doivent être visibles. Ce constat est inhérent au caractère non robuste des moindres carrés. Hager et Belhumeur proposent un M-estimateur ([12] eq49) pour remédier à cet inconvénient. C'est également cette voie qui est empruntée par Black et Jepson qui définissent une version « robuste » de l'hypothèse de conservation des niveaux de gris ([3] eq13).

Conclusion

Nous avons présenté une synthèse de différents algorithmes de suivi en temps réel pour des motifs plans. Les résultats démontrent sans conteste l'intérêt de cette phase d'intégration, tant sur des images de synthèse que sur des images réelles. Cependant, le choix des paramètres des méthodes est délicat à cause de leur nombre et de leur interdépendance.

Cet algorithme est en cours d'adaptation pour réaliser, au rythme de la vidéo, le recalage des images capturées par une caméra qui filme une scène avec une image panoramique de référence de ladite scène. Nous sommes confrontés à de nouveaux problèmes tels que :

– objets non plans et/ou non rigides ;
– objets occultés ;
– absence locale de texture.

Annexe

Échantillonnage et dérivation d'images discrètes

Soit une image discrète I. On cherche à l'« étendre » pour des coordonnées non entières (x, y) en $\widehat{I}$. On cherche aussi à calculer une dérivée $\widehat{I}'$. Plusieurs techniques existent pour ce faire. En voici trois, classées par ordre de robustesse croissante, en notant $(u, v) = (\lfloor x \rfloor, \lfloor y \rfloor)$ les coordonnées pixel sur I.

Interpolation bilinéaire.

$$\widehat{I}(x,y) = \begin{array}{l}(1+u-x)(1+v-y)I(u,v)+\\(x-u)(1+v-y)I(u+1,v)+\\(1+u-x)(y-v)I(u,v+1)+\\(x-u)(y-v)I(u+1,v+1)\end{array}$$

Cette expression est continue et dérivable par morceaux, mais elle est sensible au bruit.

Méthode brute. $\widehat{I}(x,y) = I(u,v)$. C'est la méthode la plus simple et la plus rapide. Elle ne donne pas une fonction continue et nécessite des différences finies (par exemple centrées) pour calculer sa « dérivée » (en général $d = 1$) :

$$\widehat{I}'(x,y) = \left[\begin{array}{c}\frac{\widehat{I}(u+d,v)-\widehat{I}(u-d,v)}{2d}\\ \frac{\widehat{I}(u,v+d)-\widehat{I}(u,v-d)}{2d}\end{array}\right]$$

Par convolution[17]. $\widehat{I}(x,y) = (I \otimes \Gamma)(u,v)$, où Γ est une matrice de terme général

$$\exp\left(-\frac{i^2+j^2}{2\pi\sigma^2}\right)$$

où $i,j(= -k..k)$. Cette expression est beaucoup plus résistante au bruit, mais elle n'est pas continue et nécessite des différences finies pour calculer sa « dérivée » (avec d du même ordre de grandeur que σ). Notons que $\widehat{I}(u,v)$ diffère généralement de $I(u,v)$ et que les paramètres σ et k sont à définir.

Remerciements Nous sommes reconnaissants à F. Jurie qui a répondu favorablement à notre sollicitation en vue d'approfondir la comparaison avec la méthode dont il est co-auteur. Les investigations seront menées sur la plateforme expérimentale propre à F. Jurie et M. Dhome.

Références

[1] J. R. Bergen et P. Anandan et K. J. Hanna et R. Hingorani. Hierarchical model-based motion estimation. In *ECCV'92*, pages 237–252. Springer-Verlag, May 1992.

[2] M. J. Black et P. Anandan. The robust estimation of multiple motions : parametric and piecewise-smooth flow fields. *CVIU*, 63(1) :75–104, 1996.

[3] M. J. Black et A. Jepson. Eigentracking : robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1) :63–84, 1998.

[4] A. Bartoli, N. Dalal, R. Horaud, *Motion Panoramas*, Technical Report 4771, INRIA, March 2003.

[5] A. Cavallaro et O. Steiger et T. Ebrahimi. Multiple video object tracking in complex scenes. In *Actes de ACM Multimedia*, pages 523–532, 2002.

[6] P. G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson, 1985.

[7] R. D. DeGroat et E. M. Dowling. The data least squares problem and channel equalization. *IEEE Transactions on signal processing*, 41 :407–411, 1993.

[8] J. E. Dennis et R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, Englewood Cliffs, NJ, 1983.

[9] M. Douze et B. Thiesse et V. Charvillat. Des mosaïques plus universelles, plus précises. In *Actes de RFIA*, pages 597–606, 2002.

[10] M. Douze, V. Charvillat, et B. Thiesse. Comparaison et intégration de trois algorithmes de suivi de motifs plans. *Actes de ORASIS*, pages 221–230, 2003.

[11] G. J. Edwards et T. F. Cootes et C. J. Taylor. Face recognition using active appearance models. In *European Conference on Computer Vision*, pages 581–695, 1998.

[12] G. D. Hager et P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE pami*, 20(10) :1025–1039, 1998.

[13] R. Hartley et A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, London, 2001.

[14] P. Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986.

[15] F. Jurie et M. Dhome. Un algorithme de « template matching » simple et efficace. In *Actes de RFIA*, pages 59–65, 2002.

[16] P. Lemmerling et S. Van Huffel. *Structured Least Squares : Analisys, Algorithms and Applications in Total Least Squares and Errors-in-Variables Modeling*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[17] D. Marr. *Vision*. Freeman & Co., New York, 1982.

[18] L. Masson, F. Jurie, et M. Dhome. Suivi de motifs texturés : approche hybride texture/contours. *Actes de ORASIS*, pages 231–238, 2003.

[19] J. More et B. Garbow et K. Hillstrom. User guide for minpack-1. Technical Report ANL–80–74, Argonne National Laboratory, March 1980.

[20] C. C. Paige et Z. Strakoš. *Unifying Least Squares, Total Least Squares and Data Least Squares* in *Total Least Squares and Errors-in-Variables Modeling (S. V. Huffel, P. Lemmerling ed.)*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[21] B. D. Rao. Unified Treatment of LS, TLS, and Truncated SVD Methods Using a Weighted TLS Framework. *Proceedings of the Second International Workshop on Total Least Squares and Errors-in-Variables Modeling*, pages 11–20, 1997.

[22] H. S. Sawhney et S. Ayer. Compact representation of videos through dominant and multiple motion estimation. *IEEE pami*, 18(8) :814–830, 1996.

[23] C. Schmid et R. Mohr et C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2) :151–172, 2000.

[24] J. Shi et C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, 1994.

[25] H.-Y. Shum et R. Szeliski. Panoramic image mosaicing. Technical Report MSR-TR-97-23, Microsoft Research, September 1997.

[26] S. Van Huffel et J. Vanderwalle. *The Total Least Squares Problem : Computational Aspects and Analysis*. SIAM, Philadelphia, 1991.

[27] `http://www.enseeiht.fr/lima/sigma/suiveur/suivi_2.avi`