

Formulation bidirectionnelle pour la conception d’algorithmes d’alignement d’images.

Bi-directional framework for the design of image alignment algorithms.

JB. Authesserre

R. Mégret

Y. Berthoumieu

IMS Bordeaux, UMR CNRS 5218, Université de Bordeaux

351, cours de la libération F-33405 Talence

jean-baptiste.authesserre@ims-bordeaux.fr

Résumé

Dans cet article, nous présentons et exploitons le cadre théorique bidirectionnel pour l’alignement d’images, à partir duquel nous proposons une grille de conception pour la spécification d’algorithmes d’alignement d’images. Cette méthodologie vise à rendre explicite les différents choix effectués pour concevoir un algorithme d’alignement. Elle est illustrée par la proposition d’une nouvelle approche, Asymétrique Compositionnelle, qui étend un certain nombre d’algorithmes existants. L’évaluation expérimentale montre l’avantage de cette approche en présence d’images fortement bruitées.

Mots Clef

Alignement d’images. Approches directes, inverses, symétriques, bidirectionnelles. Approche asymétrique.

Abstract

In this paper, the generic bi-directional framework for parametric image alignment is presented and used to propose a methodology to derive image alignment algorithms in a principled manner. The purpose of this methodology is to make explicit the choices required to design an algorithm. We finally propose a new approach, Asymmetric Compositional, which is shown to be a generalization of previous state-of-the art gradient-based methods. An experimental evaluation is provided to show the interest of the new approach in presence of noisy images.

Keywords

Image alignment. Forwards, inverse, symmetric and bidirectional approaches. Asymmetric approaches.

1 Introduction

L’estimation de mouvement est une tâche fondamentale de nombreuses applications en vision, notamment dans le suivi d’objets, la compression vidéo, la réalité augmentée, ou encore la création d’images mosaïque. Parmi les approches utilisées, le template matching permet de déter-

miner les paramètres du mouvement qui transforme une image en une autre. L’optimum est traditionnellement obtenu par une approche itérative qui vise à minimiser la différence pixel à pixel entre une image de référence (le template) et l’image compensée avec l’estimation courante des paramètres de mouvement.

Depuis l’algorithme de Lucas et Kanade [3], de nombreuses approches ont été proposées. Baker et Matthews ont présenté une première classification des méthodes dans [3]. Cette dernière fut étendue dans [10] pour tenir compte des derniers algorithmes proposés [5] [8], qui sont plus robustes et ont de meilleures propriétés de convergence.

Dans cet article, nous reprenons ce formalisme afin de proposer une approche de la conception d’algorithmes d’alignement d’images qui identifie clairement les différentes étapes et choix que le concepteur doit effectuer. Ce formalisme est illustré par une nouvelle famille d’algorithmes d’alignement basée gradients, nommée Asymétrique compositionnelle, qui étend plusieurs approches existantes [3] [5] [12].

Ce papier est organisé comme suit : dans la section 2, le problème de l’alignement d’images est défini, puis le formalisme bi-directionnel est présenté. Sur cette base, dans la section 3, la méthodologie de conception est alors proposée. Dans la section 4, nous détaillons l’utilisation de l’algorithme Gauss-Newton au sein de ce cadre. Dans la section 5, l’approche Asymétrique Compositionnelle est présentée et son intérêt est illustré expérimentalement dans les situations d’images fortement bruitées.

2 Cadre formel

Dans cette partie, nous formalisons le problème de l’alignement d’images, et présentons le cadre théorique bi-directionnel [10] qui justifie la méthodologie de conception proposée.

2.1 L'alignement d'images

L'alignement d'images entre un template T et une image I consiste à trouver les paramètres $\bar{\mu} \in \mathcal{P}$ tels que :

$$\forall \mathbf{x} \in R_{\text{ref}} \quad I(W(\bar{\mu}, \mathbf{x})) = T(W(\mu_{\text{ref}}, \mathbf{x})) \quad (1)$$

$\bar{\mu}$ représente les paramètres de la transformation entre l'image I et un repère de référence, et μ_{ref} représente une transformation fixe entre l'image template T et un repère de référence. R_{ref} correspond à la région d'intérêt, les positions spatiales \mathbf{x} étant exprimées dans le repère de référence.

Dans la suite, nous supposons que le modèle de mouvement W forme une action de groupe basée sur le groupe des paramètres \mathcal{P} :

$$W : (\mathcal{P} \times \mathbb{R}^2) \mapsto \mathbb{R}^2 \quad (\mu, \mathbf{x}) \mapsto W(\mu, \mathbf{x}) \quad (2)$$

avec les propriétés :

$$\forall \mathbf{x} \in \mathbb{R}^2 \quad W(\mathbf{0}, \mathbf{x}) = \mathbf{x} \quad (3)$$

$$\begin{aligned} \forall (\mu_1, \mu_2) \in \mathcal{P}^2, \forall \mathbf{x} \in \mathbb{R}^2, \\ W(\mu_1 \circ \mu_2, \mathbf{x}) = W(\mu_1, W(\mu_2, \mathbf{x})) \quad (4) \\ \mathbf{y} = W(\mu, \mathbf{x}) \text{ ssi } \mathbf{x} = W(\mu^{-1}, \mathbf{y}) \end{aligned}$$

En considérant les deux images comme des fonctions continues selon les coordonnées $\mathbf{x} \in R_{\text{ref}}$, le changement de variable $\mathbf{x} = W(\mu_{\text{ref}}^{-1} \circ \mu_T, \mathbf{z})$ utilisant un μ_T arbitraire conduit à une formulation plus générique du problème d'alignement :

$$\forall \mathbf{z} \in R \quad I(W(\mu_I, \mathbf{z})) = T(W(\mu_T, \mathbf{z})) \quad (5)$$

où $R = W(\mu_T^{-1} \circ \mu_{\text{ref}}, R_{\text{ref}})$ est la région d'intérêt transformée et

$$\mu_I = \bar{\mu} \circ \mu_{\text{ref}}^{-1} \circ \mu_T \quad (6)$$

Aligner deux images revient donc théoriquement à trouver (μ_I, μ_T) tels que (5) soit vérifiée. Une fois les paramètres bidirectionnels calculés, l'estimation $\hat{\mu}$ de la transformation directe équivalente $\bar{\mu}$ est obtenue en appliquant la loi de mise à jour issue de (6) :

$$\hat{\mu} \leftarrow \mu_I \circ \mu_T^{-1} \circ \mu_{\text{ref}} \quad (7)$$

Afin d'estimer les paramètres bidirectionnels, une métrique est construite pour comparer les parties droites et gauches de (5). Considérons le vecteur d'erreur $\mathbf{e}(\mu_I, \mu_T)$ obtenu en concaténant les différences pixel à pixel \mathbf{e}_i entre les images compensées sur un échantillonnage $(\mathbf{x}_i)_{i=1..N}$ de la région support R :

$$\mathbf{e}(\mu_I, \mu_T) = I(W(\mu_I, R)) - T(W(\mu_T, R)) \quad (8)$$

En utilisant la norme L_2 de \mathbf{e} , la fonction erreur bidirectionnelle correspond à :

$$E(\mu_I, \mu_T) = \sum_{\mathbf{z}_i \in R} \left(I(W(\mu_I, \mathbf{z}_i)) - T(W(\mu_T, \mathbf{z}_i)) \right)^2 \quad (9)$$

2.2 Catégorisation des approches

Paramétrisation incrémentale (PI). Dans la suite, μ_0 représente un estimé initial de $\bar{\mu}$, et μ_{ref} le vecteur de paramètres du mouvement entre l'image template et le repère de référence. Etant donné ces estimés initiaux, la paramétrisation incrémentale consiste à exprimer la relation fonctionnelle exprimant les paramètres bidirectionnels en fonction de vecteurs de correction $\delta\mu_I$ et $\delta\mu_T$:

$$\mu_I = \mu_I(\mu_0, \delta\mu_I) \quad \text{et} \quad \mu_T = \mu_T(\mu_{\text{ref}}, \delta\mu_T) \quad (10)$$

Lorsque la correction est nulle ($\delta\mu_I = \mathbf{0}$ et $\delta\mu_T = \mathbf{0}$), cette paramétrisation incrémentale doit vérifier :

$$\mu_I(\mu_0, \mathbf{0}) = \mu_0 \quad \text{et} \quad \mu_T(\mu_{\text{ref}}, \mathbf{0}) = \mu_{\text{ref}} \quad (11)$$

On peut différencier trois classes principales d'approches :

Directe (D). L'image est alignée sur un template qui reste fixe :

$$\mu_I = \mu_I(\mu_0, \delta\mu) \quad \text{et} \quad \mu_T = \mu_{\text{ref}} \quad (12)$$

Inverse (I). Le template est aligné sur une image qui reste fixe :

$$\mu_I = \mu_0 \quad \text{et} \quad \mu_T = \mu_T(\mu_{\text{ref}}, \delta\mu) \quad (13)$$

Bidirectionnelle. L'image et le template sont alignés l'un vers l'autre en utilisant la relation (10) complète. L'approche bidirectionnelle peut être encore décomposée en fonction des contraintes que l'on impose aux corrections $\delta\mu_I$ et $\delta\mu_T$. En particulier, les approches **Symétriques (S)** supposent que les corrections dues à $\delta\mu_I$ et $\delta\mu_T$ sont symétriques l'une de l'autre. Nous introduirons une approche **Asymétrique (A)** dans la section 5 qui se fonde sur une contrainte plus générique.

L'approche purement bidirectionnelle (B) considère que les deux corrections sont complètement indépendantes, et peuvent donc être concaténées dans un seul vecteur de paramètres $(\delta\mu_I, \delta\mu_T) \in \mathcal{P}^2$.

Par rapport aux catégories définies ci-dessus, il est encore possible de différencier les approches additives (par exemple $\mu_I = \mu_0 + \delta\mu_I$), ou compositionnelles (par exemple $\mu_I = \mu_0 \circ \delta\mu_I$). Les différentes possibilités utilisées dans des approches existantes sont introduites sous le nom de méta-paramétrisation dans [10], qui étendent la dichotomie Directe/Inverse proposée initialement par Baker et Matthews [3], en intégrant dans le même cadre formel les approches Symétriques et Bidirectionnelles proposées par Keller et Averbuch [8].

Reparamétrisation locale (RL). D'après les équations (5) et (10), aligner les 2 images revient à déterminer $\delta\mu_I$ et $\delta\mu_T$. La reparamétrisation locale consiste à exprimer ces incréments en fonction d'un vecteur unique $\delta\mathbf{v}$:

$$\delta\mu_I = \delta\mu_I(\delta\mathbf{v}) \quad \text{et} \quad \delta\mu_T = \delta\mu_T(\delta\mathbf{v}) \quad (14)$$

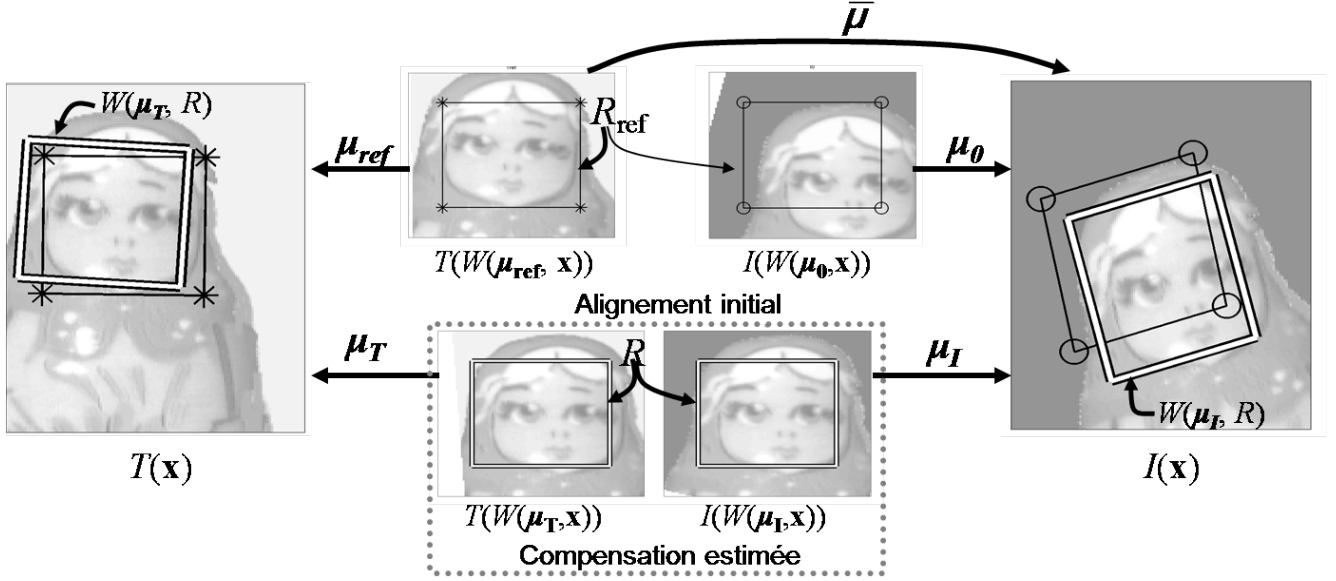


FIGURE 1 – Principe du cadre bidirectionnel pour aligner deux images I et T . Les paramètres initiaux μ_0 et μ_{ref} sont mis en parallèle avec les paramètres μ_I et μ_T conduisant à l’alignement correct. La région support R apparaît dans le repère de compensation commun $T(W(\mu_T, \mathbf{x}))$ et $I(W(\mu_I, \mathbf{x}))$. Les régions correspondantes apparaissent dans les images $T(\mathbf{x})$ et $I(\mathbf{x})$. Pour l’approche Directe, $\mu_T = \mu_{\text{ref}}$. Pour l’approche Inverse, $\mu_I = \mu_0$. Dans le cas général (approches Bidirectionnelles), μ_I et μ_T varient pendant l’optimisation.

Deux possibilités, la reparamétrisation locale directe (RLD) et la reparamétrisation locale de Lie (RLL) seront présentées à la section 3, et exploitées à la section 4.

3 Méthodologie de conception pour l’alignement d’images

Dans cette partie, nous proposons de mettre en oeuvre le formalisme proposé sous la forme d’une technique méthodique pour la conception d’un algorithme d’alignement d’images. Notre contribution par rapport à [3] et [10] est de clairement mettre en évidence les liens entre la paramétrisation du mouvement, la paramétrisation incrémentale et la reparamétrisation locale.

Choix 1	Fixer le modèle de transformation
Choix 2	Choisir la paramétrisation
Choix 3	Définir la fonction d’erreur $E(\mu_I, \mu_T)$
Choix 4	Spécifier la nature de la Paramétrisation Incrémentale (PI)
Choix 5	Définir l’espace de Reparamétrisation Locale (RL)
Choix 6	Décider de la méthode d’optimisation

TABLE 1 – Les étapes de conception d’un algorithme d’alignement d’images

Choix 1 : Fixer le modèle de transformation c’est définir la nature du mouvement que l’on cherche à estimer. Le mouvement homographique étant à la fois très couram-

ment utilisé, et pouvant être mis en oeuvre de nombreuses façons différentes, il sera choisit pour illustrer le propos.

Choix 2 : Choisir la paramétrisation c’est définir l’espace \mathcal{P} dans lequel résident les paramètres du mouvement μ_I et μ_T . Une fois la paramétrisation choisie, il est possible de définir clairement l’expression de $W(\mu, \mathbf{x})$. Dans le cas des homographies, définies par une matrice \mathbf{H} , on peut paramétriser ces dernières de nombreuses manières, notamment :

- en utilisant une matrice 3×3 dépendant de 9 paramètres réels avec une contrainte sur le déterminant $\det(\mathbf{H}) = 1$ [5], ou bien en imposant $\mu_{33} = 1$ [3] ou bien encore $\|\mathbf{H}\|_F = 1$ où $\|\cdot\|_F$ désigne la norme de Froebenius :

$$\mathbf{H}(\mu) = \begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{21} & \mu_{22} & \mu_{23} \\ \mu_{31} & \mu_{32} & \mu_{33} \end{pmatrix} = [\mu] \quad (15)$$

où $[\mu]$ désigne la réorganisation du vecteur μ sous la forme d’une matrice.

- en utilisant une paramétrisation 4 points [1] : les coordonnées de 4 points variables dans l’image I censés correspondre à 4 points fixes du template T .

Pour chaque approche, des contraintes supplémentaires peuvent être ajoutées, soit pour spécifier l’ensemble des paramètres admissibles (la paramétrisation 4 points doit conduire à la formation d’un quadrilatère non dégénéré), soit pour effectuer une estimation d’un mouvement plus contraint. Par exemple, en fixant $\mu_{31} = 0$ et $\mu_{32} = 0$ dans

l'équation (15), on contraint le mouvement à une déformation affine.

Afin d'appliquer un modèle déformable dans le cadre bidirectionnel, les propriétés de composition et d'inversion compatibles avec (2) sont nécessaires afin de pouvoir utiliser (7). Une heuristique efficace pour définir ces opérations est présentée dans [6].

Choix 3 : Définir la fonctionnelle d'erreur $E(\mu_I, \mu_T)$ dépend grandement de l'application et des performances visées : une simple norme L2 dans [3], une métrique robuste dans [2]. Il est à noter que la métrique d'erreur n'est pas nécessairement une erreur pixel à pixel, mais peut prendre une forme plus complexe. Par exemple, dans [11], l'erreur est fondée sur la dissimilarité entre distributions de couleurs locales après compensation de mouvement.

Il apparaît ainsi qu'une grande liberté est laissée au concepteur dans le choix de la métrique à cette étape pour la conception d'algorithmes plus robustes ou plus performants.

Choix 4 : Spécifier la Paramétrisation Incrementale (PI), c'est choisir l'expression des fonctions $\mu_I = \mu_I(\mu_0, \delta\mu_I)$ et $\mu_T = \mu_T(\mu_{\text{ref}}, \delta\mu_T)$. C'est donc essentiellement choisir une classe parmi (D), (I), et (B). Pour faire ce choix, l'utilisateur peut prendre en considération les éléments suivants. Sauf quand cela est spécifié (approche Inverse), une optimisation itérative par gradients est considérée :

- Les approches actuellement existantes au sein d'une classe donnée ont été démontrées équivalentes à l'ordre 1 pour une paramétrisation fixée [10].
- Une approche (I) possède des itérations rapides grâce à la possibilité de précalculer la matrice jacobienne [3]. Cette catégorie peut également utiliser un apprentissage hors ligne [7] qui permet d'estimer directement l'incrément $\delta\mu$ en fonction de l'erreur e mesurée sans passer par des approximations locales, assurant ainsi une convergence plus rapide et plus large.
- Les approches (S) conduisent souvent à une convergence en moins d'itérations (voir Fig 2) que les approches (D) et (I), surtout lorsque (1) est vérifiée. La robustesse est également améliorée [10][5][8][13].
- D'après [8], une approche (B) est plus performante qu'une approche (S) lorsque (1) n'est pas vérifiée. Nos expérimentations sur données synthétiques n'ont pas montré en pratique de différences de performances significatives entre les méthodes (B) et (S). L'étude plus approfondie de la méthode (B) dans le cadre compositionnel est l'objet de futurs travaux.

Choix 5 : Définir une reparamétrisation locale (RL), c'est spécifier l'équation (14). Ceci nécessite notamment de définir le lien entre l'espace \mathcal{P} dans lequel évoluent les incréments $\delta\mu_I$ et $\delta\mu_T$ et l'espace des paramètres où évoluent les δv_I et δv_T . On parlera de reparamétrisation locale directe (RLD), lorsque les espaces de $\delta\mu$ et δv sont confondus.

Il est également possible de recourir par exemple aux algèbres de Lie [5] [4]. En effet aux alentours de la transformation identité $W(\mathbf{0}, \cdot)$, la transformation peut-être reparamétrisée dans l'espace tangent par un vecteur δv^{lie} .

$$\begin{aligned} \mathbf{H}(\delta\mu(\delta v^{lie})) &= \expm\left(\sum_i \delta v_i^{lie} \mathbf{A}_i\right) \\ &= \expm([\mathbf{A}\delta v^{lie}]) \end{aligned} \quad (16)$$

où \expm correspond à l'exponentielle de matrice et les \mathbf{A}_i constituent une base de l'espace tangent au groupe des transformations autour de l'identité. Ces dernières peuvent être représentées de manière compacte par le biais d'une matrice \mathbf{A} dont les colonnes sont constituées des coefficients des matrices \mathbf{A}_i . Les δv_i^{lie} sont les coordonnées du vecteur δv^{lie} . La nature des \mathbf{A}_i et donc de \mathbf{A} dépend de la paramétrisation envisagée pour le groupe des transformations. On parlera alors de reparamétrisation locale de Lie (RLL).

Le formalisme de Lie permet de gérer naturellement les cas où l'espace des paramètres \mathcal{P} possède plus de dimensions que la variété sur laquelle évoluent les paramètres autorisés [4][5] comme c'est le cas avec la paramétrisation (15) avec la contrainte $\det(\mathbf{H}) = 1$.

A l'issue de la (PI) et de la (RL), la fonctionnelle d'erreur dépend uniquement du paramètre vectoriel δv , à travers les expressions (10) et (14). Ces deux choix déterminent également de manière unique la règle de mise à jour de l'estimé $\hat{\mu}$ en fonction de δv par l'équation (7).

Choix 6 : La méthode d'optimisation est la méthode utilisée pour minimiser $E(\delta v)$, la fonction erreur associée à (9). Ceci peut être réalisé en utilisant des approches basées gradients, comme les méthodes Gauss-Newton (GN) et Newton (N) [3]. Une autre possibilité est d'utiliser une approche par apprentissage, comme par exemple l'apprentissage d'un estimateur linéaire (LE) [7] [4]. Dans le paragraphe suivant, nous donnons un exemple d'optimisation utilisant l'algorithme GN. Soulignons ici le fait que la méthode d'optimisation à utiliser est indépendante a priori de la paramétrisation incrémentale, ou bien de la reparamétrisation locale.

4 Estimation itérative des paramètres bidirectionnels

Dans cette partie nous dérivons les équations clés à utiliser dans le cadre de l'utilisation de la méthode d'optimisation de type Gauss-Newton (GN), en mettant en évidence l'influence de la reparamétrisation locale. Nous illustrons ensuite ces résultats à l'aide de quelques approches classiques.

4.1 L'algorithme Gauss-Newton

L'optimisation par GN de l'erreur de fonction bidirectionnelle (9) conduit à :

$$\delta v = -(\mathbf{J}^t \mathbf{J})^{-1} \mathbf{J}^t e(\mu_0, \mu_{\text{ref}}) \quad (17)$$

où $\mathbf{J} = \left. \frac{\partial \mathbf{e}(\boldsymbol{\mu}_I(\boldsymbol{\mu}_0, \boldsymbol{\delta}\boldsymbol{\mu}_I(\boldsymbol{\delta}\mathbf{v})), \boldsymbol{\mu}_T(\boldsymbol{\mu}_{\text{ref}}, \boldsymbol{\delta}\boldsymbol{\mu}_T(\boldsymbol{\delta}\mathbf{v})))}{\partial \boldsymbol{\delta}\mathbf{v}} \right|_{\boldsymbol{\delta}\mathbf{v}=0}$ correspond à la matrice jacobienne du vecteur d'erreur \mathbf{e} défini par (8). Les fonctions considérées $I(\mathbf{x})$, $T(\mathbf{x})$, $W(\boldsymbol{\mu}, \mathbf{x})$, $\boldsymbol{\mu}_I(\boldsymbol{\mu}_0, \boldsymbol{\delta}\boldsymbol{\mu})$ et $\boldsymbol{\mu}_T(\boldsymbol{\mu}_{\text{ref}}, \boldsymbol{\delta}\boldsymbol{\mu})$ sont supposées différentiables selon les variables \mathbf{x} et $\boldsymbol{\delta}\boldsymbol{\mu}$. Les fonctions $\boldsymbol{\delta}\boldsymbol{\mu}_I(\boldsymbol{\delta}\mathbf{v})$ et $\boldsymbol{\delta}\boldsymbol{\mu}_T(\boldsymbol{\delta}\mathbf{v})$ sont supposées différentiables par rapport à $\boldsymbol{\delta}\mathbf{v}$.

La matrice jacobienne \mathbf{J} dépend de la paramétrisation incrémentale considérée. Elle peut être obtenue par la concaténation des gradients $\mathbf{J}(\mathbf{x}_i)$ correspondant aux erreurs pixel à pixel \mathbf{e}_i , où :

$$\mathbf{J}(\mathbf{x}_i) = \left. \frac{\partial \mathbf{e}_i}{\partial \boldsymbol{\delta}\mathbf{v}} \right|_{\boldsymbol{\delta}\mathbf{v}=0} = \mathbf{J}_I(\mathbf{x}_i) - \mathbf{J}_T(\mathbf{x}_i) \quad (18)$$

avec :

$$\mathbf{J}_I(\mathbf{x}_i) = \left. \frac{\partial I(W(\boldsymbol{\mu}_I(\boldsymbol{\mu}_0, \boldsymbol{\delta}\boldsymbol{\mu}_I), \mathbf{x}_i))}{\partial \boldsymbol{\delta}\boldsymbol{\mu}_I} \right|_0 \left. \frac{\partial \boldsymbol{\delta}\boldsymbol{\mu}_I(\boldsymbol{\delta}\mathbf{v})}{\partial \boldsymbol{\delta}\mathbf{v}} \right|_0 \quad (19)$$

$$\mathbf{J}_T(\mathbf{x}_i) = \left. \frac{\partial T(W(\boldsymbol{\mu}_T(\boldsymbol{\mu}_{\text{ref}}, \boldsymbol{\delta}\boldsymbol{\mu}_T), \mathbf{x}_i))}{\partial \boldsymbol{\delta}\boldsymbol{\mu}_T} \right|_0 \left. \frac{\partial \boldsymbol{\delta}\boldsymbol{\mu}_T(\boldsymbol{\delta}\mathbf{v})}{\partial \boldsymbol{\delta}\mathbf{v}} \right|_0 \quad (20)$$

Dans le cas d'erreurs entre distributions de couleurs, voir [11].

Une fois les paramètres $\boldsymbol{\delta}\mathbf{v}$ estimés, l'équation (14) permet de calculer $\boldsymbol{\delta}\boldsymbol{\mu}_I$ et $\boldsymbol{\delta}\boldsymbol{\mu}_T$. Les paramètres $\boldsymbol{\mu}_I$ et $\boldsymbol{\mu}_T$ sont alors mis à jour en tenant compte de la paramétrisation incrémentale choisie (équation (10)). La transformation directe équivalente est obtenue grâce à (6).

En particulier, dans le cas où l'on utilise une (RLD), nous avons $\boldsymbol{\delta}\boldsymbol{\mu} = \boldsymbol{\delta}\mathbf{v}$ et les équations (19) et (20) se ramènent aux expressions classiques exploitées par la plupart des auteurs [3], [12]...

Pour une (RLL), les expressions génériques (19) et (20) doivent être utilisées. Cependant remarquons que nous avons dans le cas d'une paramétrisation matricielle :

$$\text{expm}([\mathbf{A}\boldsymbol{\delta}\mathbf{v}^{lie}]) = I_3 + [\mathbf{A}\boldsymbol{\delta}\mathbf{v}^{lie}] + o(\|\mathbf{A}\boldsymbol{\delta}\mathbf{v}^{lie}\|) \quad (21)$$

et donc :

$$\left. \frac{\partial \boldsymbol{\delta}\boldsymbol{\mu}_T(\boldsymbol{\delta}\mathbf{v}^{lie})}{\partial \boldsymbol{\delta}\mathbf{v}^{lie}} \right|_0 = \left. \frac{\partial \boldsymbol{\delta}\boldsymbol{\mu}_I(\boldsymbol{\delta}\mathbf{v}^{lie})}{\partial \boldsymbol{\delta}\mathbf{v}^{lie}} \right|_0 = \mathbf{A} \quad (22)$$

En règle générale, les équations précédentes sont utilisées dans un schéma itératif. Pour une itération donnée, l'initialisation de l'itération suivante est obtenue par la règle de mise à jour tirée de (7) :

$$\boldsymbol{\mu}_0^{k+1} \leftarrow \boldsymbol{\mu}_I(\boldsymbol{\mu}_0^k, \boldsymbol{\delta}\boldsymbol{\mu}_I) \circ (\boldsymbol{\mu}_T(\boldsymbol{\mu}_{\text{ref}}, \boldsymbol{\delta}\boldsymbol{\mu}_T))^{-1} \circ \boldsymbol{\mu}_{\text{ref}} \quad (23)$$

Le paramètre $\boldsymbol{\mu}_{\text{ref}}$ reste constant pour conserver la région d'intérêt fixe [10] et éviter de recalculer \mathbf{J}_T . Ce schéma est rappelé dans la table 2 et illustré pour les différentes catégories dans la figure 2.

Etape 1	Initialisation des paramètres $\boldsymbol{\mu}_0$ et $\boldsymbol{\mu}_{\text{ref}}$
Etape 2	Calcul des vecteurs incrément $\boldsymbol{\delta}\boldsymbol{\mu}_I$ et $\boldsymbol{\delta}\boldsymbol{\mu}_T$ grâce à (17) (18) (19) (20) puis (14)
Etape 3	Calcul de $\boldsymbol{\mu}_I$ et $\boldsymbol{\mu}_T$ avec (10)
Etape 4	Mise à jour de l'estimé $\hat{\boldsymbol{\mu}}$ en utilisant la loi (23)
Etape 5	Répéter les étapes 2, 3 et 4 jusqu'à convergence

TABLE 2 – Principe de l'optimisation itérative

4.2 Quelques algorithmes classiques

Les algorithmes présentés dans cette partie utilisent une optimisation de type Gauss-Newton. Les différences entre ces approches résident essentiellement dans le choix d'une paramétrisation incrémentale et d'une reparamétrisation locale.

Dans le cas d'une (RLD), on peut définir les approches Directe Compositionnelle (DC) avec $\boldsymbol{\mu}_I = \boldsymbol{\mu}_0 \circ \boldsymbol{\delta}\boldsymbol{\mu}$ et $\boldsymbol{\mu}_T = \boldsymbol{\mu}_{\text{ref}}$ [12], Inverse Compositionnelle (IC) avec $\boldsymbol{\mu}_I = \boldsymbol{\mu}_0$ et $\boldsymbol{\mu}_T = \boldsymbol{\mu}_{\text{ref}} \circ \boldsymbol{\delta}\boldsymbol{\mu}^{-1}$ [3], ce qui conduit aux matrices jacobiennes suivantes :

$$\mathbf{J}^{DC}(\mathbf{x}_i) = \left. \frac{\partial I(W(\boldsymbol{\mu}_0 \circ \boldsymbol{\delta}\boldsymbol{\mu}, \mathbf{x}_i))}{\partial \boldsymbol{\delta}\boldsymbol{\mu}} \right|_{\boldsymbol{\delta}\boldsymbol{\mu}=0} \quad (24)$$

$$\mathbf{J}^{IC}(\mathbf{x}_i) = \left. \frac{\partial T(W(\boldsymbol{\mu}_{\text{ref}} \circ \boldsymbol{\delta}\boldsymbol{\mu}, \mathbf{x}_i))}{\partial \boldsymbol{\delta}\boldsymbol{\mu}} \right|_{\boldsymbol{\delta}\boldsymbol{\mu}=0} \quad (25)$$

L'approche Symétrique Compositionnelle (SC) [10], utilise la paramétrisation incrémentale (PI) suivante : $\boldsymbol{\mu}_I = \boldsymbol{\mu}_0 \circ \boldsymbol{\delta}\boldsymbol{\mu}$ et $\boldsymbol{\mu}_T = \boldsymbol{\mu}_{\text{ref}} \circ \boldsymbol{\delta}\boldsymbol{\mu}^{-1}$ avec comme reparamétrisation locale directe : $\boldsymbol{\delta}\boldsymbol{\mu} = (1/2)\boldsymbol{\delta}\mathbf{v}$. Ici le vecteur $\boldsymbol{\delta}\mathbf{v}$ appartient à \mathcal{P} . Cette méthode utilise donc les gradients dans les deux images pour réaliser l'alignement :

$$\mathbf{J}^{SC}(\mathbf{x}_i) = \frac{1}{2}(\mathbf{J}^{DC}(\mathbf{x}_i) + \mathbf{J}^{IC}(\mathbf{x}_i)) \quad (26)$$

Cette dernière a été démontrée équivalente au premier ordre aux approches symétriques proposées dans [8].

Dans le cas d'une (RLL), les paramétrisations incrémentales sont identiques et conduisent aux matrices jacobiennes suivantes :

$$\begin{aligned} \mathbf{J}^{DCL} &= \mathbf{J}^{DC} \mathbf{A} \\ \mathbf{J}^{ICL} &= \mathbf{J}^{IC} \mathbf{A} \\ \mathbf{J}^{SCL} &= \mathbf{J}^{SC} \mathbf{A} \end{aligned} \quad (27)$$

Les expressions de \mathbf{J}^{DCL} et \mathbf{J}^{SCL} rejoignent celles utilisées respectivement sous le nom de Gauss-Newton sur groupe de Lie dans [13] et ESM (Efficient Second-order Minimization) dans [5].

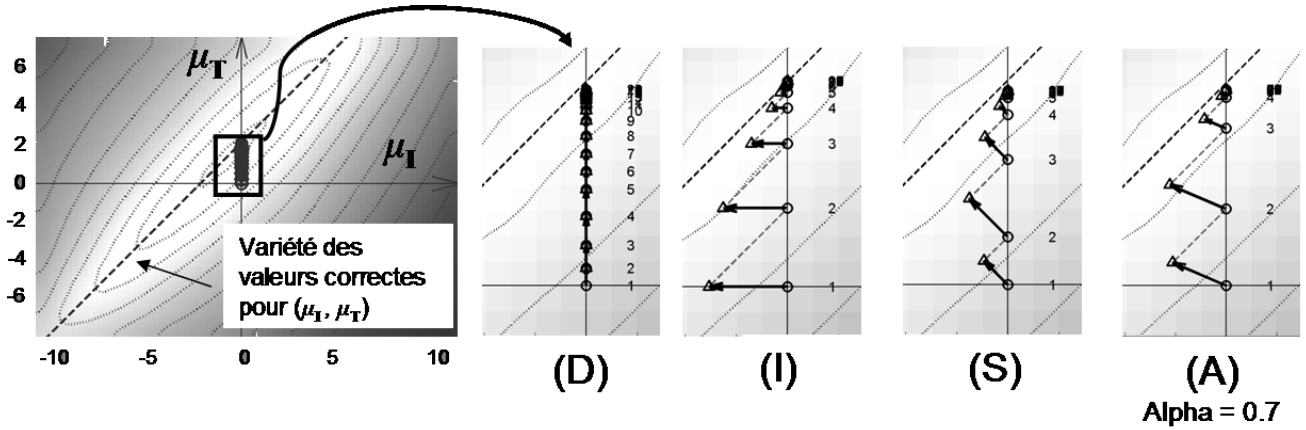


FIGURE 2 – Erreur $E(\mu_I, \mu_T)$ correspondant à l’image de la Fig. 1 affichée sur une coupe de l’espace $(\mu_{I,1}, \mu_{T,1})$ des paramètres bidirectionnels \mathcal{P}^2 , où l’index 1 représente le coefficient d’une translation horizontale. La translation est estimée en utilisant l’algorithme d’optimisation (GN) pour une méthode de chaque catégorie. Chaque itération n est tracée avec une flèche, qui relie $(\mu_0^n, \mu_{\text{ref}}^n)$ (les \odot numérotés) à (μ_I^n, μ_T^n) (les \triangle), puis par une fine ligne en pointillée qui relie (μ_I^n, μ_T^n) à l’initialisation suivante $(\mu_0^{n+1}, \mu_{\text{ref}}^{n+1})$. La vraie déformation est une translation horizontale de 2 pixels $\bar{\mu}_1 = 2$, et l’initialisation est $(\mu_{0,1}^1, \mu_{\text{ref},1}^1) = (0, 0)$. La ligne en pointillés en gras $(\mu_{I,1}, \mu_{T,1}) = (\mu_1, 2 + \mu_1)$ représente l’ensemble des estimés corrects. La forme des trajectoires reflète les contraintes imposées par la Paramétrisation Incrémentale (10) et la règle de mise à jour (23).

5 Alignement d’images asymétrique

Dans cette partie, nous proposons un nouvel algorithme d’alignement d’images afin d’illustrer la méthodologie introduite dans le chapitre précédent. Sa nouveauté réside dans le choix d’une contrainte asymétrique sur les incréments de la paramétrisation incrémentale.

5.1 Principes

La nouvelle famille d’algorithmes proposée utilise une (PI) compositionnelle :

$$\mu_I = \mu_0 \circ \delta\mu_I(\delta\mathbf{v}) \quad (28)$$

$$\mu_T = \mu_{\text{ref}} \circ (\delta\mu_T(\delta\mathbf{v}))^{-1} \quad (29)$$

Nous qualifions la nouvelle approche d’Asymétrique (A) en raison des contraintes d’asymétrie imposées sur les incréments. Les algorithmes se distinguent dans le choix de la reparamétrisation locale.

L’algorithme Asymétrique Compositionnelle (AC) utilise une reparamétrisation locale directe :

$$\delta\mu_I(\delta\mathbf{v}) = (1 - \alpha)\delta\mathbf{v} \quad (30)$$

$$\delta\mu_T(\delta\mathbf{v}) = \alpha\delta\mathbf{v} \quad (31)$$

où $\alpha \in [0, 1]$ est un paramètre réglable correspondant à une contrainte d’asymétrie imposée sur les paramètres bidirectionnels $(\delta\mu_I, \delta\mu_T)$.

L’approche Asymétrique Compositionnelle par groupe de Lie (ACL) utilise quant à elle une reparamétrisation locale (RLL), qui correspond, dans le cas de la paramétrisation des homographies par les équations (15) à :

$$[\delta\mu_I(\delta\mathbf{v}^{lie})] = \expm(\mathbf{A}(1 - \alpha)[\delta\mathbf{v}^{lie}]) \quad (32)$$

$$[\delta\mu_T(\delta\mathbf{v}^{lie})] = \expm(\mathbf{A}\alpha[\delta\mathbf{v}^{lie}]) \quad (33)$$

La contrainte d’asymétrie est exprimée dans l’algèbre de Lie au lieu d’être exprimée directement dans l’espace des paramètres \mathcal{P} .

Les matrices jacobiennes associées correspondent aux deux approches sont :

$$\mathbf{J}_\alpha^{AC} = (1 - \alpha)\mathbf{J}^{DC} + \alpha\mathbf{J}^{IC}, \quad \mathbf{J}_\alpha^{ACL} = \mathbf{J}_\alpha^{AC} \mathbf{A} \quad (34)$$

Remarquons que pour l’approche (AC), les cas particuliers $\alpha = 0$ et $\alpha = 1$ et $\alpha = 0.5$ correspondent respectivement aux approches (DC), (IC) et (SC) introduites précédemment. De façon analogue l’approche Asymétrique Compositionnelle par groupe de Lie (ACL) étend les approches (DCL), (ICL) et (SCL).

L’approche asymétrique permet ainsi l’unification des méthodes (DC), (IC) et (SC) comme étant la minimisation d’une même erreur exprimée dans un cadre bidirectionnel avec la même méthode d’optimisation (Gauss-Newton), la différence de comportement étant uniquement déterminée par la façon d’explorer cet espace bidirectionnel sous la contrainte d’asymétrie α (voir figure 2).

D’autre part, le principe de pondération des matrices jacobiennes a été étudié dans le cas de l’asservissement visuel en vitesse à partir des coordonnées de primitives spatiales détectées dans l’image [9]. Il a été ainsi montré que dans le cas de mouvement de forte amplitude, le paramètre d’asymétrie peut jouer un rôle dans la convergence de l’estimation. Dans le cas de l’alignement par différence d’images, l’amplitude est généralement plus faible du fait de l’utilisation des gradients des images qui introduisent des minima locaux. C’est pourquoi dans la suite nous nous intéressons

principalement à la robustesse de l'approche asymétrique vis à vis du bruit image.

5.2 Évaluation des performances

Dans la suite, nous proposons une évaluation expérimentale s'appuyant sur le benchmark de Baker et Matthews [3]. Les algorithmes considérés ont les spécificités algorithmiques suivantes :

- Modèle de transformation : homographie paramétrée par (15) avec la contrainte $\det(\mathbf{H}) = 1$.
- Erreur : Norme L2 (9).
- Paramétrisation Incrementale compositionnelle générique (28) (29).
- Reparamétrisation locale de Lie (RLL) avec les équations (32) et (33). Les approches Directes, Inverses et Symétriques peuvent être considérées comme des cas particuliers de l'approche Asymétrique avec les cas particuliers $\alpha = 0$, $\alpha = 1$ et $\alpha = 0,5$ pour les approches (DCL), (ICL) et (SCL) respectivement.
- Méthode d'optimisation : Gauss-Newton.

Nos expérimentations sur données synthétiques n'ont pas révélé de différences de performances entre les approches (RLD) et (RLL) dans le cadre de mouvements de faibles amplitudes étudiés ici. Pour cette raison, seuls les résultats pour la (RLL) sont montrés.

Méthode d'évaluation. Le benchmark génère des déformations spatiales correspondant à un modèle homographique en déplaçant quatre points de référence à l'aide d'une perturbation aléatoire gaussienne 2D d'écart-type σ (Point Sigma). Chaque déformation spatiale permet d'extraire d'une image de référence une imagerie template. Les algorithmes d'alignement sont ensuite utilisés pour estimer les paramètres de mouvement permettant d'aligner l'image de référence sur l'imagerie template. Nous pouvons également bruitez les images T et I avec des bruits blancs gaussiens additifs d'écart-type respectifs σ_T et σ_I . Deux critères de convergence sont envisagés :

- La fréquence moyenne de convergence : pourcentage des tests où un algorithme converge vers le bon estimé.
- La vitesse de convergence : Pour les tests pour lesquels toutes les méthodes testées ont convergé, l'erreur RMS points (distance euclidienne moyenne entre les paramétrisations 4 points de la déformation synthétique et de la déformation estimée) est tracée en fonction du nombre d'itérations.

Influence du bruit. Dans la suite, nous traçons la fréquence moyenne de convergence en fonction de Point Sigma. L'image I est altérée par un bruit aléatoire gaussien d'écart-type $\sigma_I = 25$. Le paramètre de l'algorithme (ACL) est fixé à $\alpha = 0.70$. Le choix de α sera discuté plus amplement dans la partie suivante.

Du point de vue de la fréquence moyenne de convergence (Fig. 3, en haut), l'approche (DCL) produit des résultats moins bons que l'approche (ICL), ce qui a déjà été expliqué dans [3] par le fait que les gradients de l'image I sont affectés par le bruit sur l'image. Lorsque le bruit sur l'image

n'est pas trop important, l'approche (SCL) possède une région de convergence plus grande que les approches (DCL) et (ICL) (Fig. 3 en haut à gauche), en utilisant avantageusement les gradients calculés sur les deux images I et T , comme cela a été précisé dans [5]. Lorsque le bruit image devient trop important, l'approche symétrique produit de moins bonnes performances que l'approche inverse (Fig. 3 à droite). La nouvelle approche (ACL) produit des résultats encore meilleurs, en tenant compte des deux images mais en pondérant de manière appropriée le gradient associé à l'image bruitée I . Les résultats obtenus pour la vitesse de convergence (Fig. 3, en bas) conduisent également à constater la supériorité de l'algorithme (ACL) dans cette situation.

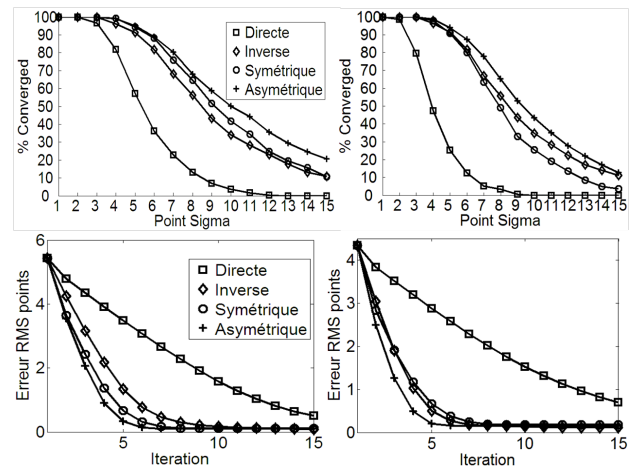


FIGURE 3 – Fréquence de convergence moyenne (en haut) et vitesse de convergence moyenne (en bas) pour les méthodes (DCL), (ICL), (SCL) et (ACL) avec $\alpha = 0.7$. A gauche, $\sigma_I = 25$. A droite, $\sigma_I = 35$.

Influence du paramètre α . Les résultats de la figure 4 sont obtenus dans les mêmes conditions que ceux des figures 3, mais en restreignant la valeur de Point Sigma à $\sigma = 8$, et en faisant varier α de 0 à 1. Pour la vitesse de convergence (Fig 4, en bas), tous les points sur une même verticale représentent l'erreur RMS points pour un α fixé, mais pour 5 itérations successives (seules les 5 premières itérations sont affichées).

Les performances des approches (DCL), (ICL) et (SCL) obtenues sur les figures 3 (à gauche) peuvent être retrouvées sur les deux graphiques de gauche de la figure 4 pour $\alpha = 0$, $\alpha = 1$ et $\alpha = 0.5$ respectivement. Il est intéressant de noter que la valeur optimale de α pour une telle configuration correspond à $\alpha = 0.7$.

Lorsqu'il n'y a aucun bruit dans les images $\sigma_I = \sigma_T = 0$ (voir fig 4, à droite), la valeur optimale est $\alpha = 0.5$, ce qui replace dans une perspective plus large les résultats théoriques de [5] et [8], qui démontrent que lorsque les deux images à aligner sont identiques au problème d'alignement près, l'utilisation d'une approche symétrique conduit à de meilleures performances.

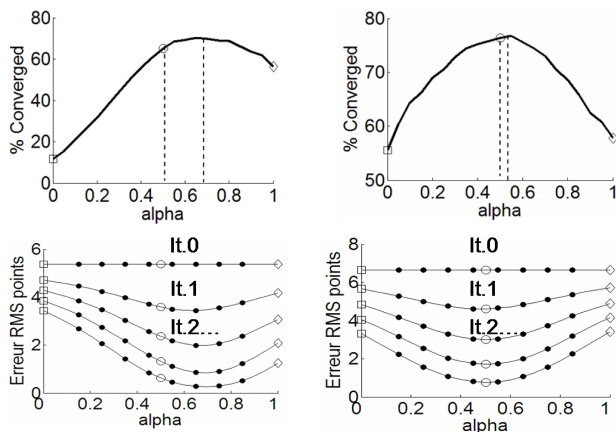


FIGURE 4 – Fréquence de convergence moyenne (en haut) et vitesse de convergence moyenne pour les 5 premières itérations (en bas) pour deux variances de bruit différentes. A gauche, $\sigma_I = 25$. A droite, $\sigma_I = 0$.

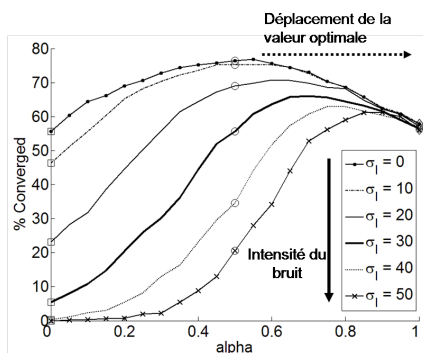


FIGURE 5 – Fréquence de convergence moyenne en fonction du paramètre α , pour une intensité de bruit croissante dans l' image I ($\sigma_T = 0$).

En revanche lorsque du bruit est présent dans l'une des images, la valeur optimale de α se déplace pour naturellement compenser les gradients affectés par le bruit. Ce déplacement est illustré dans la figure 5, où l'on peut voir que les performances se dégradent du côté de $\alpha = 0$, où les gradients sont moins pondérés. La valeur optimale de α est obtenue par un compromis entre l'utilisation des gradients des deux images et la qualité des gradients associés.

6 Conclusion

Dans cet article, nous exploitons le cadre théorique bidirectionnel pour l'alignement d'images présenté dans [10] afin de mettre en place une méthodologie pour la conception d'algorithmes d'alignement d'images. Une nouvelle approche algorithmique, qui étend un certain nombre d'approches existantes permet d'illustrer l'intérêt de cette méthodologie. L'évaluation expérimentale montre l'avantage du nouvel algorithme en présence d'images fortement bruitées. Des perspectives à ce travail concernent, au niveau de la méthode Asymétrique proposée, le réglage automatisé du paramètre α optimal. Au niveau du formalisme, il nous

semble également possible d'intégrer à celui-ci des transformations d'images autres que le mouvement (illumination, flou...) afin d'étendre cette grille de conception à une classe plus large d'algorithmes.

Références

- [1] S. Baker, A. Datta, and T. Kanade. Parameterizing homographies. Technical Report CMU-RI-TR-06-11, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 2006.
- [2] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. Lucas-Kanade 20 years on : A unifying framework : Part 2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, February 2003.
- [3] S. Baker and I. Matthews. Lucas-Kanade 20 years on : A unifying framework. *International Journal of Computer Vision*, 56(3) :221–255, 2004.
- [4] E. Bayro-Corrochano and J. Ortegon-Aguilar. Lie algebra approach for tracking and 3d motion estimation using monocular vision. *Image and Vision Computing*, 25 :907–921, 2007.
- [5] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IROS'04*, volume 1, pages 943–948, Sendai, Japan, 2004.
- [6] V. Gay-Bellile, A. Bartoli, and P. Sayd. Feature-driven direct non-rigid image registration. In *BMVC'07*, 2007.
- [7] F. Jurie and M. Dhome. Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :996–1000, 2002.
- [8] Y. Keller and A. Averbuch. Fast motion estimation using bi-directional gradient methods. *IEEE Transactions on Image Processing*, 13(8) :1042–1054, August 2004.
- [9] M. Marey and F. Chaumette. Analysis of classical and new image-based control laws. In *RFIA*, 2008.
- [10] R. Mégret, J-B Authesserre, and Y. Berthoumieu. The bi-directional framework for unifying parametric image alignment approaches. In *ECCV08*, pages 400–411, 2008.
- [11] R. Mégret, M. Mikram, and Y. Berthoumieu. Inverse composition for multi-kernel tracking. In *ICVGIP*, pages 480–491, 2006.
- [12] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2) :101–130, 2000.
- [13] T. Vercauteren, X. Pennec, E. Malis, A. Perchant, and N. Ayache. Insight into efficient image registration techniques and the demons algorithm. In *IPMI*, volume 4584, pages 495–506, 2007.