International Journal of Computer Vision (2015)
(accepted for publication) DOI: 10.1007/s11263-014-0769-6

Rationalizing Efficient Compositional Image Alignment

The Constant Jacobian Gauss-Newton optimization algorithm

Enrique Muñoz · Pablo Márquez-Neila · Luis Baumela

Received: date / Accepted: date

Abstract We study the issue of computational efficiency for Gauss-Newton (GN) non-linear least-squares optimization in the context of image alignment. We introduce the *Constant Jacobian Gauss-Newton* (CJGN) optimization, a GN scheme with constant Jacobian and Hessian matrices, and the *equivalence* and *independence* conditions as the necessary requirements that any function of residuals must satisfy to be optimized with this efficient approach. We prove that the *Inverse Compositional* (IC) image alignment algorithm is an instance of a CJGN scheme and formally derive the *compositional* and *extended brightness constancy* assumptions as the necessary requirements that must be satisfied by any image alignment problem so it can be solved with an efficient compositional scheme. Moreover, in contradiction with previous results, we also prove that the forward and inverse compositional algorithms are not equivalent. They are equivalent, however, when the *extended brightness constancy* assumption is satisfied. To analyze the impact of the satisfaction of these requirements we introduce a new image alignment evaluation framework and the concepts of *short-* and *wide*-baseline Jacobian. In wide-baseline Jacobian problems the optimization will diverge if the requirements are not satisfied. However, with a good initialization, a short-baseline Jacobian problem may converge even if the requirements are not satisfied.

Keywords Image Registration · Tracking · Efficient Gauss-Newton Optimization · Efficient Compositional Image Alignment

Enrique Muñoz · Pablo Márquez-Neila · Luis Baumela
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid
E-mail: lbaumela@fi.upm.es

1 Introduction

Direct image alignment approaches are widely employed computer vision techniques. They can be used for densely and efficiently estimating the motion of low or smooth textured surfaces, like human faces [1,9,11,13,15,17, 23,24,27,29,31,33,34], medical image registration [14], augmented reality [21] and vision-based tracking and servoing [7,10], to name a few. The fast local convergence of the Gauss-Newton (GN) non-linear least-squares make it the most popular optimization technique for image alignment. This approach, however, is computationally intensive given the typically large number of residuals that emerge in these optimizations.

In the last decade the *compositional image alignment* approach [2,3,8,21,14,25,32] has received much attention. The most remarkable feature of this approach is that, under the IC formulation introduced by Baker and Matthews [2,3], the Hessian and Jacobian matrices in the GN optimization are constant. Thus, it provides a very efficient image alignment algorithm adequate, for example, for tracking in real-time with a mobile phone [20]. Compositional schemes have been used also for efficient and robust face image alignment [28,33], for fitting 2D [23,1,15] and 3D [24,26,31] deformable face models, model-based 2D [21,3] and 3D [12] rigid target tracking as well as medical [14], 2D [5] and 3D [34] geometric and photometric image registration.

Despite the widespread use of GN techniques in computer vision and the success of the efficient IC image alignment approach, we still do not have a theory about what makes efficiency afforded in GN optimization. Moreover, the IC algorithm lacks a formal presentation. The relation between IC and the standard GN optimization scheme is not well understood. We need clear explanation to the fact that we may be able move in the domain of parameters of a non-trivial optimization function and have a constant descent direction. Traditionally, this has received an intuitive explanation based on what has been called the *compositional trick* [5]: if in the optimization we exchange the role of the incoming image and the template, then we may be able to compute the descent direction in terms of the latter, that is constant [2,3]. However, this trick cannot be considered a general explanation. Hence, we need to know why and under what requirements some GN optimizations, and the associated image alignment problems, can be solved with constant Jacobian and Hessian matrices.

In this paper we provide a theory about efficiency, in terms of constant Jacobian and Hessian matrices, in GN optimization, as we revisit the problem of image alignment. In Section 3 we introduce the concept of *equivalent* and *independent* parameters for residuals with two sets of parameters. We then prove that any residual function with equivalent and independent parameters can be optimized with a GN procedure featuring constant Jacobian and Hessian matrices. We call this approach the *Constant Jacobian Gauss-Newton* (CJGN) optimization procedure. Further, in Section 4 we introduce the *compositional* and the *extended brightness constancy* assumptions. We prove that if they are satisfied, we can approximate the residuals of the compositional image alignment approach in terms of a function with equivalent and independent parameters, thus, leading to the *Efficient Compositional* (EC) image alignment algorithm, that features constant Jacobian and Hessian matrices. We also prove that, under the same requirements, IC is also an instance of a CJGN scheme and that IC and EC are actually the same algorithm, being EC a formal derivation of IC within the CJGN scheme. Moreover, we also prove that forward and inverse compositional image alignment approaches are equivalent when the *extended brightness constancy* assumption is satisfied. This is in contradiction with previous results that proved their unconditional equivalence [3]. To further clarify this point we also provide a counter example proving that, in general, forward and inverse compositional approaches cannot be equivalent. In Section 5 we experimentally study the impact that the satisfaction of the above mentioned requirements has in the performance of the efficient compositional schemes. To this end, we extend the traditional concept of short and wide-baseline image alignment to that of short and wide-baseline Jacobian and introduce a new image alignment evaluation framework that considers the wide-baseline Jacobian setting. Finally, in Sections 6 and 7 we discuss the results in the paper and draw conclusions.

1.1 Notation

Regular lowercase symbols such as x, f, represent scalar values or functions; bold symbols such as $\boldsymbol{\theta}$, $\mathbf{r}$ are vectors; teletype uppercase symbols such as $\mathtt{R}$ or $\mathtt{H}$ indicate matrices; uppercase calligraphic symbols as $\mathcal{X}$ are sets.

If $f(x)$ is the application of f on $x \in \mathcal{X}$, $\mathbf{f}(\mathcal{X})$ is the vector resulting from stacking the result of the application of f to each element in $\mathcal{X}$.

We have chosen regular capital letters (I, T) to refer to gray-scale images. An image is an application $I : \mathbb{R}^d \rightarrow \mathbb{R}$, where d is the dimension of the image domain. In our case $d = 2$. Since we work with image sequences we add a second parameter, t, $I : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$, that represents time or the order of the image in the sequence.

The models that we use in the paper are defined in in 2D, $\mathbb{P}^2$, and 3D, $\mathbb{R}^3$. Hence, to relate a point with its gray value in an image we need to project it onto $\mathbb{R}^2$. We denote with $p : \mathcal{D} \rightarrow \mathbb{R}^2$, $\mathcal{D} \in \{\mathbb{R}^3, \mathbb{P}^2\}$, the projection function. Thus, the gray value of point $\mathbf{x}$ in the image T is given by $T(p(\mathbf{x}))$. We use the following notation to reduce verbosity $I(p(\mathbf{x}), t) \equiv I[\mathbf{x}, t]$ and $T(p(\mathbf{x})) \equiv T[\mathbf{x}]$.

2 Background and related work

In this section we review the related literature and introduce basic background concepts about image alignment.

2.1 Image alignment and the BCA

Image alignment is the process of bringing into coincidence the projection of a target in two or more images. The target model is represented as a *template*, T, and provides information such as the structure and texture of the object to be aligned.

Direct image alignment approaches are based on the *Brightness Constancy Assumption* (BCA).

Definition 1 (Brightness Constancy Assumption) *Given a template T and a warping function $\mathbf{f}$, for any image I at time t there exists a parameter vector $\boldsymbol{\mu}$ such that*

$$I[\mathbf{f}(\mathbf{x}, \boldsymbol{\mu}), t] = T[\mathbf{x}], \; \forall \mathbf{x} \in \mathcal{V}, \tag{1}$$

where $\mathbf{f}(\mathbf{x}, \boldsymbol{\mu}) : \mathcal{D} \times \mathbb{R}^p \rightarrow \mathcal{D}$ is the warping function that models changes in the visual appearance of the moving object transforming points in $\mathcal{V} \in \mathcal{D}$ into $\mathcal{V}' \in \mathcal{D}$ (see Fig. 1).

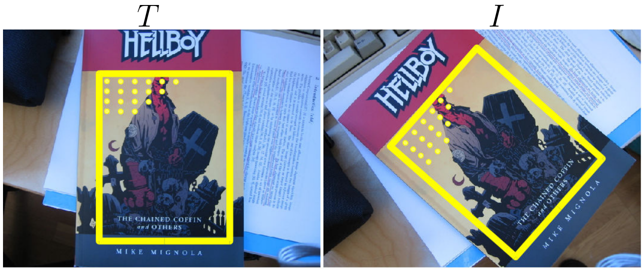

Fig. 1 Domains $\mathcal{D}$, $\mathcal{V}$, $\mathcal{X}$ involved in an image alignment problem. $\mathcal{D}$ depends on the domain in which the warping function is defined. Typical domains for this example would be $\mathbb{R}^2$ or $\mathbb{P}^2$. (**left**) We overlay the target region $\mathcal{V}$ (*yellow square*) onto the template T. The support set $\mathcal{X}$ for the target region $\mathcal{V}$ is given by the set of yellow dots. (**right**) We transform $\mathcal{V}$ and $\mathcal{X}$ into $\mathcal{V}'$ and $\mathcal{X}'$ by means of $\mathbf{f}$.

In practice, the alignment process is solved as an optimization

$$\boldsymbol{\mu}_t = \arg\min_{\boldsymbol{\mu}} \frac{1}{2}\sum_{\mathbf{x}\in\mathcal{X}} (I[\mathbf{f}(\mathbf{x},\boldsymbol{\mu}),t] - T[\mathbf{x}])^2, \tag{2}$$

where $\mathcal{X}$ is a finite subset of $\mathcal{V}$, the *support set* (see Fig. 1). We use as error measure the *Sum of Squared Differences* [3,17,21]. However, our results also apply to other robust measures such as those in [16,17,31].

The `GN` approach [30] minimizes (2) iteratively by solving at each step a local optimization $\arg\min_{\Delta\boldsymbol{\phi}} \mathcal{L}(\Delta\boldsymbol{\phi})$,

$$\mathcal{L}(\Delta\boldsymbol{\phi}) = \frac{1}{2}\sum_{i=1}^{N} r_i(\boldsymbol{\mu},\Delta\boldsymbol{\phi})^2 = \frac{1}{2}\mathbf{r}^\top(\boldsymbol{\mu},\Delta\boldsymbol{\phi})\mathbf{r}(\boldsymbol{\mu},\Delta\boldsymbol{\phi}) \tag{3}$$

where $N = \mathrm{card}(\mathcal{X})$ and $\mathbf{r}(\boldsymbol{\mu},\Delta\boldsymbol{\phi})$ is the vector of registration residuals,

$$\mathbf{r}(\boldsymbol{\mu},\Delta\boldsymbol{\phi}) = I[\mathbf{f}(\mathcal{X},\mathbf{w}(\boldsymbol{\mu},\Delta\boldsymbol{\phi})),t+1] - T[\mathcal{X}], \tag{4}$$

and $\boldsymbol{\mu} \leftarrow \mathbf{w}(\boldsymbol{\mu},\Delta\boldsymbol{\phi})$, $\mathbf{w}:\mathbb{R}^p\times\mathbb{R}^q\rightarrow\mathbb{R}^p$, is the step update function.

A feature common to most image registration procedures is the existence of two vectors of warping parameters. Vector $\boldsymbol{\mu}$ is the *global parameter* vector. It represents the motion from the origin of the space of warping parameters up to the point where the target is located at time t. Vector $\Delta\boldsymbol{\phi}$ is the *local parameter* vector, that represents an incremental motion of the target with respect to $\boldsymbol{\mu}$. We have used a different notation for each vector to denote the fact that we may use a local warping function different from the global one. In Section 3 we will see that the existence of two spaces of parameters will be the key for the efficiency of the optimization algorithm, since, in certain circumstances, we will be able to compute the optimization step for any $\boldsymbol{\mu}$ in a fixed point of the space of local parameters.

Depending on the analytic formulation of the vector of registration residuals and on the step update function, several image alignment algorithms have emerged in the literature. In the following we review those directly related to the efficient compositional image alignment problem.

2.2 Non-efficient image alignment

The standard approach for solving (3) is based on using an additive step update function, $\mathbf{w}(\boldsymbol{\mu},\Delta\boldsymbol{\phi}) \equiv \boldsymbol{\mu}+\Delta\boldsymbol{\phi}$, following the fundamental structure of a descent optimization method [30]. It is usually known as “Lucas and Kanade registration” (`LK`) [3], in acknowledgment of the work in which this type of registration was first introduced [22].

The *forward compositional* (`FC`) approach has drawn the attention of many researchers because in many image motion models (e.g. homographies) the composition of two warps is more natural than their addition. In this approach we allow the composition of two different warp functions, $\mathbf{f},\mathbf{g}:\mathcal{D}\times\mathbb{R}^{\{p,q\}}\rightarrow\mathcal{D}$, that share the same domain, but may have different definitions. The step update function is then defined as $\boldsymbol{\mu}\leftarrow\boldsymbol{\mu}' = \mathbf{w}(\boldsymbol{\mu},\Delta\boldsymbol{\phi})$ when $\mathbf{f}(\mathbf{g}(\cdot,\Delta\boldsymbol{\phi}),\boldsymbol{\mu}) = \mathbf{f}(\cdot,\boldsymbol{\mu}')$.

Rewriting the residuals (4) in terms of the compositional step update, yields the `FC` image residuals

$$\mathbf{r}_c(\boldsymbol{\mu},\Delta\boldsymbol{\phi}) = I[\mathbf{f}(\mathbf{g}(\mathcal{X},\Delta\boldsymbol{\phi}),\boldsymbol{\mu}),t+1] - T[\mathcal{X}]. \tag{5}$$

This is a slight generalization to the standard approach [3] in which $\mathbf{f}$ and $\mathbf{g}$ are the same function.

A important drawback of both the additive and compositional approaches is their high computational cost, caused by the typically large set of image residuals to be optimized.

2.3 Inverse Compositional image alignment

The `IC` image alignment algorithm was introduced by Baker and Matthews [2,3] to alleviate the cost of the `FC` and `LK` image alignment approaches. The key to the efficiency of the algorithm is switching the role of the image and the template in (5), and apply what Bartoli calls the *compositional trick* [5] to obtain a new set of residuals

$$\mathbf{r}_{ic}(\boldsymbol{\mu},\Delta\boldsymbol{\phi}) = I[\mathbf{f}(\mathcal{X},\boldsymbol{\mu}),t+1] - T[\mathbf{h}(\mathcal{X},\Delta\boldsymbol{\phi})], \tag{6}$$

where $\mathbf{h}:\mathcal{D}\times\mathbb{R}^q\rightarrow\mathcal{D}$ is a warping function such that $\mathbf{g}\equiv\mathbf{h}^{-1}$. In this case, the minimum of (3) is given by

$$\Delta\boldsymbol{\phi} = -(\mathtt{H}^{ic})^{-1}(\mathtt{J}^{ic})^\top\mathbf{e}(\boldsymbol{\mu}), \tag{7}$$

where $\mathtt{J}^{ic}$ and $\mathtt{H}^{ic}$ are respectively the Jacobian and GN's Hessian matrices [30], and

$$\mathbf{e}(\boldsymbol{\mu}) = I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t+1] - T[\mathcal{X}]. \tag{8}$$

The step update is given by the composition $\mathbf{f}(\mathbf{x}, \boldsymbol{\mu}) \leftarrow \mathbf{f}(\mathbf{h}^{-1}(\mathbf{x}, \Delta\boldsymbol{\phi}), \boldsymbol{\mu})$.

The most remarkable feature of this approach is that neither $\mathtt{J}^{ic}$ nor $\mathtt{H}^{ic}$ depend on $\boldsymbol{\mu}$. Thus they are constant during the minimization and may be computed off-line (see Algorithms 1 and 2).

Although IC is a widely used alignment algorithm [12, 20, 23, 24, 26, 29, 31, 33, 34] its relation with the standard additive GN optimization is not well understood. For this reason sometimes it is used in contexts in which it has poor accuracy [26, 12] or in applications in which it loses much of its efficiency since $\mathtt{J}^{ic}$ and $\mathtt{H}^{ic}$ have to be updated during the optimization [34] or computed off-line [31]. In Sections 3 and 4 we formally establish this relation.

The IC residuals in (6) are different from the compositional residuals in (5). So, the minimization (3) is different in each algorithm. Hence, they are different alignment approaches. In Section 4.3 we will further discuss this result and the conditions under which FC and IC algorithms are equivalent.

Algorithm 1 Inverse compositional algorithm (off-line stage)

```
Compute template gradient : $\nabla\mathtt{T} \leftarrow \left.\frac{\partial T[\mathbf{x}]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathcal{X}}$
Evaluate $g$-warp Jacobian: $\mathtt{G} \leftarrow \left.\frac{\partial \mathbf{g}(\mathcal{X}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0}$
Compute the constant Jacobian $\mathtt{J}^{ic} \leftarrow \nabla\mathtt{T}\,\mathtt{G}$
Compute GN Hessian $\mathtt{H}^{ic} = (\mathtt{J}^{ic})^\top \mathtt{J}^{ic}$
Compute the pseudo-inverse $(\mathtt{J}^{ic})^+ = (\mathtt{H}^{ic})^{-1}(\mathtt{J}^{ic})^\top$
```

Algorithm 2 Inverse compositional algorithm for image I_{t+1} knowing an initial estimation $\boldsymbol{\mu}$ (on-line stage)

```
repeat
    Compute $\mathbf{e}(\boldsymbol{\mu})$ from (8)
    Compute local parameters $\Delta\boldsymbol{\phi} = -(\mathtt{J}^{ic})^+\mathbf{e}(\boldsymbol{\mu})$
    Update global parameters $\boldsymbol{\mu}$:
    $\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}) \leftarrow \mathbf{f}(\mathbf{h}^{-1}(\mathcal{X}, \Delta\boldsymbol{\phi}), \boldsymbol{\mu})$
until $\|\Delta\boldsymbol{\phi}\| < \epsilon$
```

3 The Constant Jacobian Gauss-Newton method

In this section we will formally prove that for a special type of residual functions, $\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi})$, defined over two sets of *equivalent* and *independent* parameters, we can introduce a GN optimization scheme with constant Jacobian and Hessian matrices. We call this new optimization scheme the *Constant Jacobian Gauss-Newton* optimization algorithm (CJGN). In the next section we will use this general scheme as a framework in which we can formally derive efficient compositional image alignment algorithms.

Definition 2 (Equivalent parameters) *Given a function* $\mathbf{r} : D_{\boldsymbol{\theta}} \times D_{\boldsymbol{\phi}} \to \mathbb{R}^n$, *we say that* $\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi})$ *has* equivalent parameters *with a given* pivot value $\boldsymbol{\phi}_0$ *if* $\forall \boldsymbol{\theta}' \in D_{\boldsymbol{\theta}}$, $\forall \boldsymbol{\phi}' \in D_{\boldsymbol{\phi}}$, $\exists \boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\theta}', \boldsymbol{\phi}', \boldsymbol{\phi}_0) \in D_{\boldsymbol{\theta}}$ *such that* $\|\mathbf{r}(\boldsymbol{\theta}', \boldsymbol{\phi}')\|_2 = \|\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi}_0)\|_2$.

Note that a sufficient condition for having equivalent parameters is $\mathbf{r}(\boldsymbol{\theta}', \boldsymbol{\phi}') = \mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi}_0)$.

Example 1
1. *The function* $r : \mathbb{R}^2 \to \mathbb{R}$, $r(\theta, \phi) = \theta + \phi$ *has equivalent parameters for any pivot value. For example, consider* $\phi_0 = 1$. *Then, for any* $\phi', \theta' \in \mathbb{R}$, *one can find* $\theta = \theta' + \phi' - 1$ *such that* $\theta' + \phi' = \theta + \phi_0$.
2. *the function* $r : \mathbb{R}^2 \to \mathbb{R}$, $r(\theta, \phi) = \theta \cdot \phi$ *has equivalent parameters for any non-zero pivot value* $\phi_0 \neq 0$.
3. *The function* $r : \mathbb{R}^2 \to \mathbb{R}$, $r(\theta, \phi) = \theta^2 + \phi^2$ *has equivalent parameters only for the pivot value* $\phi_0 = 0$.

Functions with equivalent parameters define an equivalence relation $\sim$ that groups all values of $D_{\boldsymbol{\theta}} \times D_{\boldsymbol{\phi}}$ that lead to equivalent values of $\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi})$. $D_{\boldsymbol{\theta}} \times D_{\boldsymbol{\phi}} / \sim$ is the quotient set made up of all equivalence classes given by $\sim$. Each equivalence class has the canonical representation $(\boldsymbol{\theta}, \boldsymbol{\phi}_0)$.

Example 2 $r(\theta, \phi) = \theta + \phi$ *with pivot value* $\boldsymbol{\phi}_0 = 0$ *has an equivalence class made up of all pairs* (θ, ϕ) *whose evaluation is* 0: $(1, -1) \sim (2, -2) \sim (\theta, -\theta)$. *Its canonical representation is* $(0, 0)$.

Another equivalence class is made up of all pairs whose evaluation is 1: $(2, -1) \sim (3, -2) \sim (\theta, -\theta + 1)$. *Its canonical representation is* $(1, 0)$.

Equivalent parameters are specially useful in the GN method when the criterion function to optimize has the form $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \|\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi}_0)\|_2^2$, where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}_0$ are equivalent sets of parameters. Given an initial guess $\boldsymbol{\theta}_k$, the standard GN procedure computes the Jacobian $\mathtt{J}(\boldsymbol{\theta}_k, \boldsymbol{\phi}_0)$ as the derivative of $\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to $\boldsymbol{\theta}$, to get the increment $\Delta\boldsymbol{\theta}$ to $\boldsymbol{\theta}_k$ [30]. However, thanks to the equivalence of parameters, we can proceed in a slightly different manner. We evaluate the Jacobian as the derivative of $\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to the auxiliary set of parameters $\boldsymbol{\phi}$, and compute the the increment $\Delta\boldsymbol{\phi}$. This leads to a new partial solution $(\boldsymbol{\theta}_k, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi})$. Expressing this value in its canonical form $(\boldsymbol{\theta}_k, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi}) \sim (\boldsymbol{\theta}_{k+1}, \boldsymbol{\phi}_0)$

we obtain the updated value $\boldsymbol{\theta}_{k+1}$. This procedure is repeated until convergence or until a stopping criterion is reached.

The Jacobian $\mathtt{J}(\boldsymbol{\theta}_k, \boldsymbol{\phi}_0) = \frac{\partial \mathbf{r}(\boldsymbol{\theta}_k,\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\Big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0}$ is always evaluated at $\boldsymbol{\phi}_0$, but $\boldsymbol{\theta}_k$ is different at each step. Therefore, in general, it is not constant. However, in some cases this Jacobian does not depend on $\boldsymbol{\theta}$. In these cases, we say that $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are *independent.*

Definition 3 (Independent parameters) *Given a function* $\mathbf{r} : D_{\boldsymbol{\theta}} \times D_{\boldsymbol{\phi}} \rightarrow \mathbb{R}^n$*, we say that* $\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi})$ *has* independent parameters*, or that* $\boldsymbol{\theta}$ *and* $\boldsymbol{\phi}$ *are independent if* $\frac{\partial \mathbf{r}(\boldsymbol{\theta},\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\Big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \mathtt{J}(\boldsymbol{\phi}_0) \quad \forall \boldsymbol{\theta} \in D_{\boldsymbol{\theta}}$*, where* $\mathtt{J}(\boldsymbol{\phi}_0)$ *is a constant Jacobian matrix that depends on the pivot* $\boldsymbol{\phi}_0$.

Example 3
1. *Functions* $r(\theta, \phi) = \theta + \phi$ *and* $r(\theta, \phi) = \theta^2 + \phi^2$ *have independent parameters.*
2. *Function* $r(\theta, \phi) = \theta \cdot \phi$ *has equivalent but not independent parameters.*

Since the Jacobian does not depend on $\boldsymbol{\theta}$ under the independent parameters assumption, we define a new Jacobian function as $\mathtt{J}_{GN'}(\boldsymbol{\phi}) = \mathtt{J}(\boldsymbol{\theta}, \boldsymbol{\phi})$. Moreover, as we stated above, the Jacobian is always evaluated at $\boldsymbol{\phi}_0$. We define the *constant Jacobian* of a function with independent and equivalent parameters as $\mathtt{J}_C = \mathtt{J}_{GN'}(\boldsymbol{\phi}_0)$. Hence, given a residual function $\mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\phi})$ with independent and equivalent parameters, one can optimize the sum of squared residuals with a constant Jacobian $\mathtt{J}_C$ using Algorithm 3.

Algorithm 3 The Constant Jacobian `GN` method.

```
Compute $\mathtt{J}_C = \mathtt{J}_{GN'}(\boldsymbol{\phi}_0)$
Compute GN Hessian $\mathtt{H}_C = \mathtt{J}_C^\top \mathtt{J}_C$
Compute the pseudo-inverse $\mathtt{J}_C^+ = \mathtt{H}_C^{-1} \mathtt{J}_C^\top$
Start with an initial guess $\boldsymbol{\theta}_0$
$k \leftarrow 0$
repeat
    $\Delta\boldsymbol{\phi}_k = -\mathtt{J}_C^+ \mathbf{r}(\boldsymbol{\theta}_k, \boldsymbol{\phi}_0)$
    Compute $\boldsymbol{\theta}_{k+1}$ such that $(\boldsymbol{\theta}_{k+1}, \boldsymbol{\phi}_0) \sim (\boldsymbol{\theta}_k, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi}_k)$
    $k \leftarrow k + 1$
until $\|\Delta\boldsymbol{\phi}\| < \epsilon$ or $k >= k_{\max}$
```

In this algorithm the parameter vector $\boldsymbol{\theta}_k$ is the equivalent of the global parameter vector in the registration algorithms described in Section 2. Similarly, $\boldsymbol{\phi}$ is the local parameter vector. The pivot point $\boldsymbol{\phi}_0$ represents an arbitrary point in the local parameter space where the constant Jacobian is computed.

3.1 Example

To illustrate the `CJGN` method, we show how it minimizes a simple residual. Consider the objective function $E(\theta) = \|r(\theta)\|^2$, $r(\theta) = \theta^2 - 3$. Analytically we can see that $E(\theta)$ reaches its minima at $\theta = \pm\sqrt{3}$. Now, we introduce an auxiliary parameter ϕ that allows us to use the `CJGN`, $r(\theta, \phi) = \theta^2 + \phi^2 - 4$, and choose an arbitrary pivot $\phi_0 = 1$. This version has also equivalent and independent parameters, and its Jacobian at $\phi = 1$ is $\mathtt{J}_C = 2$. Therefore, we aim to minimize $E(\theta) = \|r(\theta, \phi_0)\|^2$ for the initial guess $\theta_0 = 0.2$ using the constant Jacobian $\mathtt{J}_C = 2$. Table 1 shows the information obtained in the minimization. In it we can see how the algorithm converges to the minimum, $\sqrt{3}$, in a few iterations.

Let us observe how the `CJGN` works. In each iteration, it performs a standard `GN` step over the auxiliary parameter ϕ, and then it translates this step into an update of the target parameter θ, leaving the auxiliary parameter unchanged. This is possible because of the equivalence of the parameters. Moreover, thanks to the independence of the parameters, the `GN` Jacobian remains the same at the new point, and the procedure is repeated (see Fig. 2).

The upper image in Fig. 2 shows the contour lines of $\|r(\theta, \phi)\|^2$ and the behavior of the `CJGN` method in the two-dimensional space (θ, ϕ). The dashed black line indicates the pivot point of the auxiliary parameter ϕ_0. Blue points represent the values (θ_k, ϕ_0), and white points are the values $(\theta_k, \phi_0 + \Delta\phi_k)$. The numbers beside each point indicate the iteration number.

Each iteration starts at a blue point. The blue arrow shows the standard `GN` step over the auxiliary parameter. The white arrow shows how the `GN` step is translated into its canonical representation, leading to the starting (blue) point of next iteration. Level set lines represent the parameter equivalence classes. In each iteration, the blue point and the white point of the previous iteration must be on the same level set, *i.e.*, both values must belong to the same equivalence class. Moreover, all canonical representations must lie on the dashed black segment.

For example, in the first iteration, the algorithm starts at blue point $(0.2, 1)$. The parameter update gives the white point $(0.2, 2.48)$. Its canonical representation is given by the blue point $(2.2782, 1)$, and therefore it is on the same contour line as $(0.2, 2.48)$. The second iteration starts at $(2.2782, 1)$.

The bottom plot in Fig. 2 shows the cost function $E(\theta) = \|r(\theta, \phi_0)\|^2$ when the auxiliary parameter is fixed at the pivot point, that corresponds to the slice under the dashed black line in the upper image. The cost function at the pivot point is equal to the objective function $E(\theta)$, and the `CJGN` finds its minimum as desired.

Iteration	θ_k	$r(\theta_k, \phi_0)$	$\mathtt{J}_C$	$\Delta\phi_k$	$\phi_0 + \Delta\phi_k$	θ_{k+1}
0	0.2	-2.96	2	1.48	2.48	2.2782
1	2.2782	2.1904	2	-1.095	-0.9052	2.0493
2	2.0493	1.1994	2	-0.5997	0.4002	1.8329
3	1.8329	0.3597	2	-0.1798	0.8202	1.7414
4	1.7414	0.0323	2	-0.0162	0.9838	1.7321
...						
7	1.73205	$< 1 \cdot 10^{-15}$	2	$< 1 \cdot 10^{-15}$	1	1.73205

Table 1 Minimization of $E(\theta) = ||\theta^2 - 3||^2$ with the `CJGN` method. The residual function is $r(\theta, \phi) = \theta^2 + \phi^2 - 4$ with pivot value $\phi_0 = 1$ and constant Jacobian $\mathtt{J}_C = 2$.

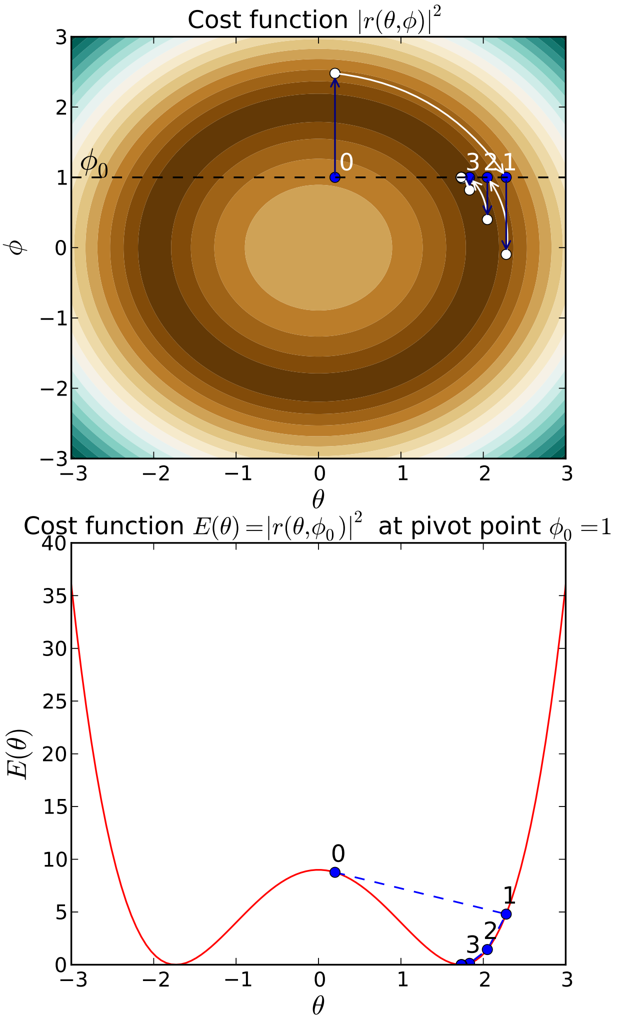

Fig. 2 Example of the `CJGN` method with the residual function $r(\theta, \phi) = \theta^2 + \phi^2 - 4$. The isocontours in the top plot represent the equivalence classes defined by $r(\theta, \phi)$.

4 Efficient compositional image alignment

In this section we revisit the problem of compositional image alignment. Our goal here is deriving an efficient compositional image alignment approach within the `CJGN` scheme. To this end we introduce the *compositional* and the *extended brightness constancy* requirements as the image alignment counterparts to the *equivalent* and *independent* parameters requirements of the `CJGN` scheme. If these requirements are satisfied then the image alignment problem can be solved using a compositional approach featuring constant Jacobian and Hessian matrices, since it is an instance of a `CJGN` scheme. Moreover, we will also prove that this efficient compositional approach, termed `EC`, is actually the `IC` algorithm. So, `EC` will not be a new compositional algorithm, but a formal derivation of `IC` within the `CJGN` framework. In this way will formally derive both `IC` and its necessary requirements.

4.1 The Efficient Compositional (`EC`) algorithm

Our goal now is to optimize expression (2) using a `CJGN` scheme. As explained in Section 3, we must first include an auxiliary parameter vector. The compositional method introduced in Section 2 gives us a way of including it, according to the compositional residual in (5). We will use the point $\boldsymbol{\phi}_0$, whose properties will be given below, as pivot in the `CJGN` scheme.

Functions $\mathbf{f}$ and $\mathbf{g}$ are image warps which may belong to two different function families. The choice of these function families is limited by the following requirement.

Requirement 1 (Compositional Assumption) *The composition* $\mathbf{f} \circ \mathbf{g}$ *must be a* f*-warp, i.e., for any* $\boldsymbol{\phi}$ *and for any* $\boldsymbol{\mu}$*, there exists a* $\boldsymbol{\mu}'$ *such that* $\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}') = \mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}), \boldsymbol{\mu})$*. Besides, there must be an identity g-warp with parameters* $\boldsymbol{\phi}_0$ *such that*

$$\mathcal{X} = \mathbf{g}(\mathcal{X}, \boldsymbol{\phi}_0). \tag{9}$$

We will denote the *Compositional Assumption* as CA.

Proposition 1 *When warps f and g satisfy the CA, $\mathbf{r}_c(\boldsymbol{\mu}, \Delta\boldsymbol{\phi})$ in* (5) *has equivalent parameters with pivot point $\boldsymbol{\phi}_0$.*

Proof From the CA, for each $\boldsymbol{\mu}$ and $\Delta\boldsymbol{\phi}$, we can find a $\boldsymbol{\mu}'$ such that

$$\begin{aligned}\mathbf{r}_c(\boldsymbol{\mu}, \Delta\boldsymbol{\phi}) &= I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \Delta\boldsymbol{\phi}), \boldsymbol{\mu}), t+1] - T[\mathcal{X}] \\ &= I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}'), t+1] - T[\mathcal{X}].\end{aligned}$$

From (9),

$$\begin{aligned}&I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}'), t+1] - T[\mathcal{X}] = \\ &\qquad = I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}_0), \boldsymbol{\mu}'), t+1] - T[\mathcal{X}] \\ &\qquad = \mathbf{r}_c(\boldsymbol{\mu}', \boldsymbol{\phi}_0).\end{aligned}$$

Subsequently, $\mathbf{r}_c(\boldsymbol{\mu}, \Delta\boldsymbol{\phi}) = \mathbf{r}_c(\boldsymbol{\mu}', \boldsymbol{\phi}_0)$. □

The CA is equivalent to the closure under composition condition introduced in [3] when $\mathbf{f}$ and $\mathbf{g}$ belong to the same function family. It is a necessary condition for any compositional image alignment procedure, regardless of whether it features a constant Jacobian or not. This is also the only requirement demanded by the `FC` algorithm.

Unfortunately, the compositional residuals in (5) do not have independent parameters. The Jacobian of $\mathbf{r}_c(\boldsymbol{\mu}, \Delta\boldsymbol{\phi})$ with respect to $\Delta\boldsymbol{\phi}$ depends on $\boldsymbol{\mu}$. We now show how to get a first order approximation to the residuals that features independent parameters. To this end, we must introduce a new requirement.

Requirement 2 *The result of an infinitesimal disturbance $\delta\boldsymbol{\phi}$ in the $(f \circ g)$-warped image with warping parameters $\boldsymbol{\mu}$ at time t must be equal to the result of the same displacement in the g-warped template image, i.e., for every $(t, \boldsymbol{\mu})$*

$$\left.\frac{\partial I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}), \boldsymbol{\mu}), t]}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \left.\frac{\partial T[\mathbf{g}(\mathcal{X}, \boldsymbol{\phi})]}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \mathtt{J}^{ec}.$$

Proposition 2 *When Requirement 2 is satisfied, the residual $\mathbf{r}_c(\boldsymbol{\mu}, \Delta\boldsymbol{\phi})$ in* (5) *can be linearly approximated by the efficient compositional residual $\mathbf{r}_{ec}(\boldsymbol{\mu}, \Delta\boldsymbol{\phi}) = \mathbf{e}(\boldsymbol{\mu}) + \mathtt{J}^{ec}\Delta\boldsymbol{\phi}$, where $\mathbf{e}(\boldsymbol{\mu})$ is given by* (8)*, that features independent parameters.*

Proof We can approximate the first term in the right hand side of (5) with a first order Taylor series expansion around $\boldsymbol{\phi}_0$ and t_τ, where t_τ is a time instant such that the brightness constancy assumption is satisfied for the motion parameters $\boldsymbol{\mu}$ reached at the iteration τ in the minimization (3), $t \leq t_\tau \leq t+1$.

$$\begin{aligned}&I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \Delta\boldsymbol{\phi}), \boldsymbol{\mu}), t+1] = \\ &\qquad I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t_\tau] \\ &\qquad + \left.\frac{\partial I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}), \boldsymbol{\mu}), t_\tau]}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} \Delta\boldsymbol{\phi} \\ &\qquad + \left.\frac{\partial I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t]}{\partial t}\right|_{t=t_\tau} (t+1-t_\tau) \\ &\qquad + o(\Delta\boldsymbol{\phi}, (t+1-t_\tau))^2.\end{aligned} \tag{10}$$

This linear approximation is valid if $\Delta\boldsymbol{\phi}$ and $(t + 1 - t_\tau)$ are small. We can then make the additional approximation

$$\begin{aligned}&\left.\frac{\partial I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t]}{\partial t}\right|_{t=t_\tau} (t+1-t_\tau) = \\ &\qquad\qquad I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t+1] - I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t_\tau] \\ &\qquad\qquad + o(t+1-t_\tau)^2\end{aligned} \tag{11}$$

From (11) and (10) we get

$$\begin{aligned}&I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \Delta\boldsymbol{\phi}), \boldsymbol{\mu}), t+1] = \\ &\qquad I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t+1] + \mathtt{J}(\boldsymbol{\mu}, t_\tau)\Delta\boldsymbol{\phi} \\ &\qquad + o(\Delta\boldsymbol{\phi}, (t+1-t_\tau))^2,\end{aligned} \tag{12}$$

where

$$\mathtt{J}(\boldsymbol{\mu}, t_\tau) = \left.\frac{\partial I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}), \boldsymbol{\mu}), t_\tau]}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0}.$$

If Requirement 2 is satisfied we can safely replace in (12) the Jacobian $\mathtt{J}(\boldsymbol{\mu}, t_\tau)$ with the constant Jacobian $\mathtt{J}^{ec}$.

We then obtain an approximation to the warped image at the iteration τ in the minimization (3) with a mapping that depends only on $\Delta\boldsymbol{\phi}$. Introducing this result in (5) and discarding the infinitesimals of order two and higher we obtain the expression for the efficient compositional residuals

$$\begin{aligned}\mathbf{r}_{ec}(\boldsymbol{\mu}, \Delta\boldsymbol{\phi}) &= I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t+1] - T[\mathcal{X}] + \mathtt{J}^{ec}\Delta\boldsymbol{\phi} \\ &= \mathbf{e}(\boldsymbol{\mu}) + \mathtt{J}^{ec}\Delta\boldsymbol{\phi},\end{aligned} \tag{13}$$

The Jacobian of $\mathbf{r}_{ec}(\boldsymbol{\mu}, \boldsymbol{\phi})$ in (13) with respect to the local parameter at the pivot point is given by

$$\left.\frac{\partial \mathbf{r}_{ec}(\boldsymbol{\mu}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \mathtt{J}^{ec},$$

that does not depend on $\boldsymbol{\mu}$. Hence, the efficient compositional residuals have independent parameters. □

To better understand this requirement we introduce the following lemma.

Lemma 1 *Let $\Delta\phi$ be a small increment in the parameters of the local warping function* $\mathbf{g}$. *If and only if Requirement 2 is satisfied, then, up to a first order approximation, the Brightness Constancy Assumption also holds for the set of points $\mathcal{X}' = \mathbf{g}(\mathcal{X}, \phi_0 + \Delta\phi)$.*

Proof The BCA for the set of points $\mathcal{X}'$ is given by

$$I[\mathbf{f}(\mathcal{X}', \boldsymbol{\mu}), t] = T[\mathcal{X}'], \tag{14}$$

that can also be expressed as

$$I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi}), \boldsymbol{\mu}), t] = T[\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi})]. \tag{15}$$

The first order Taylor series expansion of the first term in (15) is given by

$$\begin{aligned} I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi}), \boldsymbol{\mu}), t] = & \\ = I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t] & \\ + \left.\frac{\partial I[\mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}), \boldsymbol{\mu}), t]}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} & \Delta\boldsymbol{\phi} + o(\Delta\boldsymbol{\phi})^2. \end{aligned} \tag{16}$$

Similarly, the first order Taylor series expansion of the second term in (15) is given by

$$\begin{aligned} T[\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi})] = & \\ = T[\mathcal{X}] + \left.\frac{\partial T[\mathbf{g}(\mathcal{X}, \boldsymbol{\phi})]}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} & \Delta\boldsymbol{\phi} + o(\Delta\boldsymbol{\phi})^2. \end{aligned} \tag{17}$$

The equality of the first terms on the right hand side of (16) and (17) is proved by the BCA.

Proving the necessity and sufficiency is now trivial. If Requirement 2 is satisfied, then up to a first order approximation, (14) holds. Conversely, if (14) holds, then (16) and (17) must be equal, and consequently Requirement 2 must be satisfied. □

Lemma 1 provides an alternative and more intuitive expression for Requirement 2, given by (15), that together with (1) may be interpreted as an *Extended Brightness Constancy Assumption* (EBCA). The EBCA is an assumption stronger than the BCA, since it entails the equality of both image values and gradients. This is the key assumption for achieving efficiency in GN-based image alignment approaches, since the gradient of the incoming image may be expressed in terms of that of the template, that is constant.

The residuals of any image alignment problem whose warping function satisfies the CA and the EBCA have both equivalent and independent parameters. In this case, it is possible to optimize the image alignment objective function in equations (2) and (3) with an efficient GN scheme featuring constant Jacobian and Hessian matrices using the EC algorithm, an instance of the CJGN scheme introduced in Section 3. The EC algorithm is listed in Algorithms 4 and 5. In Algorithm 4 we compute the generalized inverse of the constant Jacobian matrix $\mathtt{J}^{ec}$, that may be computed off-line. Algorithm 5 optimizes (3) for the image at time $t+1$ starting at a previous known estimation of the tracking parameters $\boldsymbol{\mu}$ and using the constant Jacobian computed in the off-line stage.

Algorithm 4 Efficient compositional algorithm (off-line stage)

```
Compute template gradient : $\nabla\mathtt{T} \leftarrow \left.\frac{\partial T[\mathbf{x}]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathcal{X}}$
Evaluate $g$-warp Jacobian: $\mathtt{G} \leftarrow \left.\frac{\partial \mathbf{g}(\mathcal{X},\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0}$
Compute constant Jacobian $\mathtt{J}^{ec} = \nabla\mathtt{T}\,\mathtt{G}$
Compute the GN Hessian $\mathtt{H}^{ec} = (\mathtt{J}^{ec})^\top \mathtt{J}^{ec}$
Compute the pseudo-inverse $(\mathtt{J}^{ec})^+ = (\mathtt{H}^{ec})^{-1}(\mathtt{J}^{ec})^\top$
```

Algorithm 5 Efficient compositional algorithm for image I_{t+1} from an initial estimation $\boldsymbol{\mu}$ (on-line stage)

```
repeat
    Compute image residual $\mathbf{e}(\boldsymbol{\mu})$ from (8)
    Compute local parameters $\Delta\boldsymbol{\phi} = -(\mathtt{J}^{ec})^+\mathbf{r}(\boldsymbol{\mu}, \boldsymbol{\phi}_0)$
    Update global parameters $\boldsymbol{\mu}$: $(\boldsymbol{\mu}', \boldsymbol{\phi}_0) \sim (\boldsymbol{\mu}, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi})$,
    i.e. $\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}) \leftarrow \mathbf{f}(\mathbf{g}(\mathcal{X}, \boldsymbol{\phi}_0 + \Delta\boldsymbol{\phi}), \boldsymbol{\mu})$
until $\|\Delta\boldsymbol{\phi}\| < \epsilon$
```

4.2 On the relation between EC and IC

Any compositional image alignment scheme that satisfies the CA may be optimized using the FC algorithm. We have just proved that if it also satisfies the EBCA, then it can be optimized with an efficient algorithm using the EC approach. The IC algorithm [3] is an alternative efficient compositional approach. Here we analyze the relation between the IC and EC algorithms.

Proposition 3 *The residuals of IC,* $\mathbf{r}_{ic}$ *in* (6)*, and those of EC,* $\mathbf{r}_{ec}$ *in* (13)*, are the same, up to a first order approximation.*

Proof Assuming $\mathbf{g}^{-1} \equiv \mathbf{h}$, we can rewrite (6) as

$$\mathbf{r}_{ic}(\boldsymbol{\mu}, \Delta\boldsymbol{\mu}) = I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t+1] - T[\mathbf{g}^{-1}(\mathcal{X}, \Delta\boldsymbol{\phi})]. \tag{18}$$

Making a first order Taylor series expansion of the second term of (18) at $\boldsymbol{\phi}_0$ and using the derivative of inverse warps (see Appendix A) we get

$$T[\mathbf{g}^{-1}(\mathcal{X}, \Delta\boldsymbol{\phi})] = T[\mathcal{X}] - \left.\frac{\partial T[\mathbf{g}(\mathcal{X}, \boldsymbol{\phi})]}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} \Delta\boldsymbol{\phi} + o(\Delta\boldsymbol{\phi})^2.$$

(19)

Replacing (19) in (18) and discarding higher order terms we get the first order approximation to the `IC` residuals

$$\mathbf{r}_{ic}(\boldsymbol{\mu}, \Delta\boldsymbol{\phi}) = I[\mathbf{f}(\mathcal{X}, \boldsymbol{\mu}), t+1] - T[\mathcal{X}] + \mathtt{J}^{ec}\Delta\boldsymbol{\phi},$$

that coincides with the `EC` residuals $\mathbf{r}_{ec}$ in (13). □

So, `EC` and `IC` approaches have identical residuals, up to a first order approximation, and both feature constant Jacobian and Hessian matrices. Moreover, if we compare the `IC` algorithm 1 and 2 with `EC` 4 and 5 we can see that they are actually the same algorithm. Hence, we can consider `EC` as a formal derivation of `IC` within the `CJGN` scheme.

4.3 On the relation between `FC` and `IC`

In their seminal work, Baker and Matthews claim that `FC` and `IC` are equivalent algorithms [3]. However, we have just proved that any alignment problem satisfying the CA requirement can be solved using the `FC` algorithm, whereas to use the `IC` it must also satisfy the EBCA. So, in general, they cannot be equivalent.

Proposition 4 *If the EBCA is satisfied, then the residuals of the `FC` algorithm,* $\mathbf{r}_c$ *in* (5)*, and those of the `IC` algorithm,* $\mathbf{r}_{ic}$ *in* (6)*, are equal, up to a first order approximation.*

Proof If the EBCA is satisfied, we can rewrite the `FC` residual in (5), as

$$\mathbf{r}_c(\boldsymbol{\mu}, \Delta\boldsymbol{\phi}) = I[\mathbf{f}(\mathbf{g}(\mathcal{X}', \Delta\boldsymbol{\phi}), \boldsymbol{\mu}), t+1] - T[\mathcal{X}']. \tag{20}$$

We may then introduce the change of variables $\mathcal{X}' = \mathbf{h}(\mathcal{X}, \Delta\boldsymbol{\phi}); \mathcal{X} = \mathbf{g}(\mathcal{X}', \Delta\boldsymbol{\phi})$ in (20) to reach (6), the residuals of the `IC` algorithm. □

Hence, if the EBCA is satisfied, $\mathbf{r}_c = \mathbf{r}_{ic}$, up to a first order approximation. Thus, optimization (3) would be the same for both approaches. In this case we can make the *compositional trick* and efficiently minimize the residuals of a compositional warping function using `IC`. Hence, `FC` and `IC` are equivalent, *i.e.* perform the same steps and reach the same solution, if the EBCA is satisfied.

So, what is wrong with the *compositional trick*? It is based on the following change of variables, $\mathcal{X}' = \mathbf{g}(\mathcal{X}, \Delta\boldsymbol{\phi})$. If we introduce that change of variables in (5) we get

$$\mathbf{r}_c(\boldsymbol{\mu}, \Delta\boldsymbol{\phi}) = I[\mathbf{f}(\mathcal{X}', \boldsymbol{\mu}), t+1] - T[\mathbf{g}^{-1}(\mathcal{X}', \Delta\boldsymbol{\phi})], \tag{21}$$

that is almost like $\mathbf{r}_{ic}$ in (6), except for the domain in which it is defined, $\mathcal{X}'$. From a practical point of view (21) is useless for solving an alignment problem, since $\mathcal{X}'$ and $\Delta\boldsymbol{\phi}$ are both unknown. To obtain a useful expression, such as (6), the `IC` algorithm assumes that $\mathcal{X} \approx \mathcal{X}'$, up to a zeroth order approximation (see [3] Section 3.2.5).

It is easy to find a counterexample to the assumption $\mathcal{X} \approx \mathcal{X}'$, since, in general, $\mathcal{X} \neq \mathcal{X}'$ no matter how small is $\Delta\boldsymbol{\phi}$. Suppose that the object to be aligned is a square region on a 3D plane rotating around one axis, such that $\mathbf{x} \in \mathcal{V} \subset \mathbb{R}^3$ (see Fig.3(right)). The assumption $\mathcal{X} = \mathcal{X}'$ only holds along the rotation axis. The distance between corresponding points in $\mathcal{X}$ and $\mathcal{X}'$ is larger the further away from the axis. For any $\Delta\boldsymbol{\phi}$, we can always find points $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' = \mathbf{f}(\mathbf{x}, \Delta\boldsymbol{\phi}) \in \mathcal{X}'$ that are arbitrarily distant. So, for this simple example, $\mathcal{X} \neq \mathcal{X}'$.

Moreover, although the BCA holds for $\mathcal{X}$, this does not guarantee that it also holds for $\mathcal{X}'$. In general, $I[\mathbf{f}(\mathcal{X}', \boldsymbol{\mu}), t] \neq T[\mathcal{X}']$. Hence, we cannot replace $\mathcal{X}$ by $\mathcal{X}'$ in (21) to obtain the `IC` residuals (6). However, when the EBCA is satisfied, if the BCA holds for $\mathcal{X}$ then it must also hold for $\mathcal{X}'$. In this case we can safely replace $\mathcal{X}$ for $\mathcal{X}'$ in (21) to obtain (6).

4.4 Geometrical interpretation of the EBCA

The geometrical intuition behind the EBCA implies that a small $\mathbf{g}$-warped increment in the warping parameters of the support set, $\mathcal{X} \subset \mathcal{V}$, generates a new support set $\mathcal{X}'$ that is also contained in $\mathcal{V}$ so that both, $\mathcal{X}$ and $\mathcal{X}'$, satisfy the BCA. Intuitively, this happens for most warps when $\mathcal{V}$ is an open set, for example a 2D image in $\mathbb{R}^2$ or a 3D image stack in $\mathbb{R}^3$. Otherwise, for example in 2.5 image alignment, since $\mathcal{V}$ is a surface embedded in $\mathbb{R}^3$, the set of warps is more restrictive. A simple example of this situation is given by a square planar object sliding along itself (see Fig. 3(left)). In this case, the sets of points $\mathcal{X}$ and $\mathcal{X}'$ are in the plane containing the template. Hence, the EBCA is satisfied (see Appendix B for a proof).

4.5 `EC`/`IC` and linear regression

Alignment approaches based on linear regression are also an instance of a `CJGN` scheme. The iterative solution for the minimization of residual (6) given by (7), defines a constant linear relation between the local parameter vector, $\Delta\boldsymbol{\phi}$, and the errors (8), $\mathbf{e}(\boldsymbol{\mu})$, given by $\mathtt{A} = -(\mathtt{H}^{ic})^{-1}(\mathtt{J}^{ic})^{\top}$. Approaches to image alignment based on linear regression learn the constant linear mapping

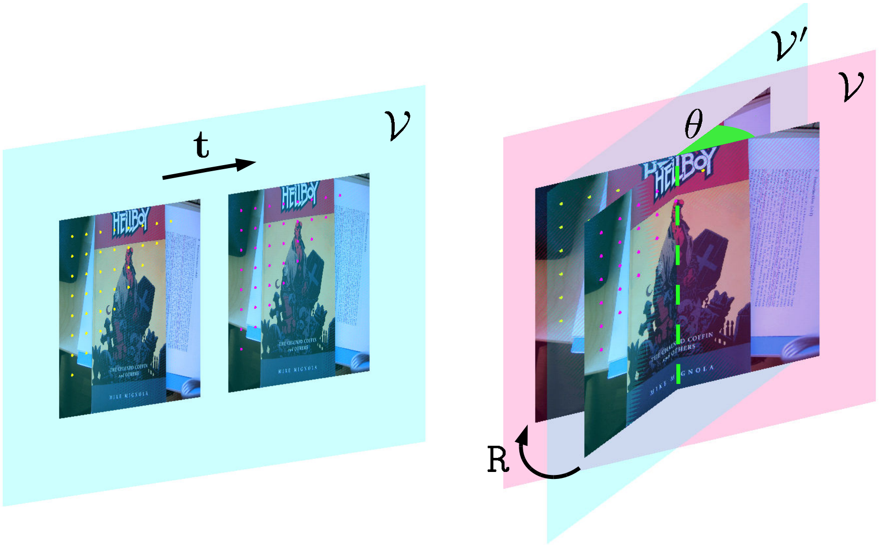

Fig. 3 Different motions of a 3D plane. (left) A 3D plane sliding along itself. The motion warp satisfies the EBCA, hence, the sets of points $\mathcal{X}$ (yellow dots) and $\mathcal{X}'$ (red dots) belong to the same domain $\mathcal{V} \subset \mathbb{R}^3$. (right) A plane rotating $\Delta\phi$ degrees. Yellow marks represent points $x \in \mathcal{X} \subset \mathcal{V}$. Red marks represent points $x' \in \mathcal{X}' \subset \mathcal{V}'$

matrix, $\mathtt{A}$, as the solution of a least-squares problem between a random set of warping parameters and its associated set of error residuals [21,35,18,20,19]. As proved in this section, this constant linear mapping is valid if the minimization problem features equivalent and independent parameters. Hence, tracking techniques based on linear regressors must also satisfy these requirements.

5 Experiments

The goal of the experiments performed in this section is evaluating efficient and non-efficient image alignment algorithms assessing the impact of the satisfaction of the CA and the EBCA in the performance. To this end, we consider of a set of alignment problems with different degrees of complexity and use optimization algorithms satisfying all or some of the requirements.

5.1 Evaluation framework

Problems dealing with the alignment of a pair of images have traditionally been grouped into short- and wide-baseline, depending on the distance between the two aligned images. Efficient image alignment techniques involve a third image in the optimization process, the *template*. It is the image where we compute the constant Jacobian used in the optimization. Hence, efficient image alignment algorithms introduce a new categorization depending on whether the template is close to the two aligned images or far from any of them. In each of these cases we say that we are respectively in a *short-* or *wide-baseline Jacobian* setting.

In a registration problem the template often coincides with one of the two aligned images (see Fig. 4(top)). Hence, it would imply a short- or wide-baseline Jacobian setting depending on whether the aligned images are short- or wide-baseline. On the other hand, tracking problems are normally considered short-baseline, since usually the distance between two consecutive images in the sequence is small. However, the efficient alignment solution entails a wide-baseline Jacobian, since the template is usually an image acquired from a very different point of view, e.g. the first image in the sequence (see Fig. 4(bottom)).

The evaluation framework used in [3] has become a standard in image alignment problems [14,8,25,33]. It uses a fixed region in the image, typically a square, as template where the Jacobian is computed. Then this region is randomly distorted and aligned back to the template. This setup models a typical registration problem that only involves short-baseline Jacobian settings. Hence, it is not adequate for evaluating tracking algorithms.

In this section we introduce a new image alignment evaluation framework that involves both short- and wide-baseline Jacobian settings.

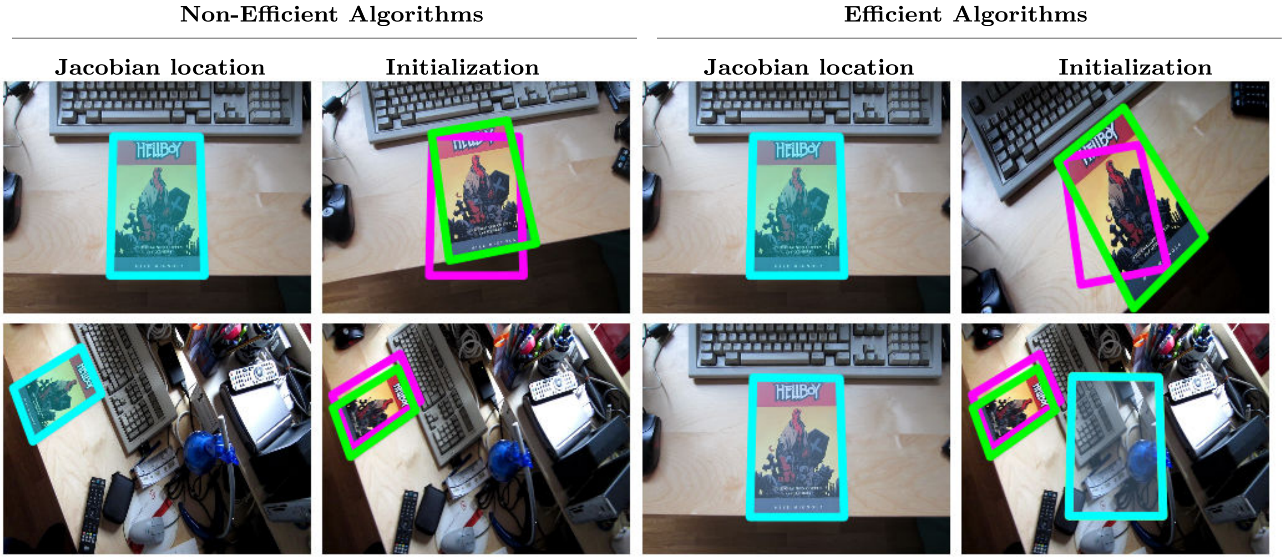

Fig. 4 Short- and Wide-baseline Jacobian. Legend:*blue*, location where the Jacobian is computed; *pink*, initial point of the optimization; and *green*, actual optimum. (Top-left) Non-efficient registration naturally uses a short-baseline Jacobian: the Jacobian is computed at the initial point of the optimization. (Bottom-left) Non-efficient tracking uses a short-baseline Jacobian: the Jacobian is computed at the previous estimation of the optimum. Efficient registration and efficient tracking compute the Jacobian at the template. (Top-Right) Efficient registration typically uses a short-baseline Jacobian. (Bottom-Right) Efficient tracking uses a wide-baseline Jacobian.

5.2 Experimental setup

We evaluate the three optimization algorithms described in Section 2, LK, FC, and IC. We proved in Section 4 that EC and IC approaches are the same algorithm. Their performance for all the experiments in this section was indistinguishable. Hence here we will only refer to IC.

We perform our tests with a synthetically generated textured plane depicting four Gaussian-based gradient patterns (see Fig. 5). This is a good texture for direct registration [6]. The size of the images is 640×480 pixels in which the textured pattern occupies a square of 272×272 pixels. In our experiments, we generate the synthetic images of a moving plane with a set of warping functions. We have selected three warping functions that satisfy some or all of the requirements in Section 4:

- **3DRT.** Points in the plane are defined in Euclidean 3D space, $\mathbf{x} \in \mathbb{R}^3$. This is the standard six-degrees-of-freedom rigid body motion warp for a planar surface in 3D, 3DRT:$\mathbb{R}^3 \rightarrow \mathbb{R}^3$. It satisfies the CA, since the composition of rigid motions is also a rigid motion. However, it does not satisfy the EBCA.
- **H6.** Here points are in 2D, $\mathbf{x} \in \mathbb{P}^2$. The motion is defined by a homography parametrized by the rotation and translation parameters representing relative camera orientation, H6:$\mathbb{P}^2 \rightarrow \mathbb{P}^2$. This function satisfies the EBCA, but since it is not closed under composition it fails to satisfy the CA. In this model we perform the parameter update as described in [12].
- **H8.** Here, points are also in 2D, $\mathbf{x} \in \mathbb{P}^2$. The motion is defined by the standard eight-degrees-of-freedom homography, H8:$\mathbb{P}^2 \rightarrow \mathbb{P}^2$. This warping function is closed under composition and satisfies both the CA and the EBCA.

In our experiments we combine an optimization algorithm —LK, FC and IC— with a warping function— H8, H6, and 3DRT. We denote it by appending the warping function name to the algorithm. We have selected the following combinations LKH8, ICH8, LKH6 and FC3DRT because they comprise additive, compositional and efficient compositional algorithms that satisfy all their requirements. Finally, we have also selected ICH6 and IC3DRT because they are efficient compositional algorithms that do not satisfy the CA and the EBCA, respectively. Table 2 summarizes the experiments.

We evaluate the performance of each experiment with a set of quantitative features:

- The *image re-projection error*,
$$\varepsilon(\hat{\boldsymbol{\mu}}) = \frac{1}{N} \|\mathbf{p}(\mathbf{f}(\mathcal{X}; \hat{\boldsymbol{\mu}})) - \mathbf{p}(\mathbf{f}(\mathcal{X}; \boldsymbol{\mu}^*))\|,$$
measures the accuracy of the results, i.e. how close are the estimated parameters $\hat{\boldsymbol{\mu}}$ from the actual optimum, $\boldsymbol{\mu}^*$.
- The *optimization time*, measured in seconds, gives us information about the efficiency.
- The *frequency of convergence* gives us information about the robustness of an algorithm.

- The *rate of convergence* informs us about how many iterations are required for the algorithm to converge to the optimum.

We evaluate the selected algorithms by registering the template to an image. All the selected optimization algorithms are based on a `GN` scheme, where the derivatives are numerically computed using central differences. The minimization starts at $\boldsymbol{\mu}_0$. It stops when it reaches 50 iterations or when the increment to the parameters is negligible, that is, $|||\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t||| \approx 0$. n

5.3 Datasets

We generate 48 000 experimental tests to evaluate the selected algorithms. Each experiment comprises an image of the textured plane with random pose, and the initial guess $\boldsymbol{\mu}_0$ for the `GN` optimization. The ground-truth parameters $\boldsymbol{\mu}^*$ represent the pose at which the textured plane is rendered.

We organize these 48 000 synthetic experiments in three datasets, DS1, DS2, and DS3, with 16 000 trials each. These datasets simulate the experimental conditions in short- and wide-baseline Jacobian problems. We choose a fixed location in the space of parameters, $\boldsymbol{\mu}_{\mathtt{J}}$, where we compute the Jacobian. For each dataset, we generate ground truth target positions $\boldsymbol{\mu}^*$ sampling from a uniform distribution with support region $\Psi \cup -\Psi$ around $\boldsymbol{\mu}_{\mathtt{J}}$, where $\Psi = (a, b)$ and $-\Psi = (-b, -a)$, and the range (a, b) is defined in Table 3. We establish the plane pose with two distributions: $\Psi_{\mathtt{R}}$ for the rotation angles, and $\Psi_{\mathbf{t}}$ for translation parameters (see Table 3). These distributions describe non-overlapping regions in the parameter space centered in $\boldsymbol{\mu}_{\mathtt{J}}$. From these values we can see that the dataset DS1 represents a short-baseline Jacobian problem. The dataset DS3 is a wide-baseline Jacobian problem. Finally, DS2 is a transition between short- and wide-baseline Jacobian problems (see Fig. 5).

We randomly compute the initial guess for the optimization as a random displacement from $\boldsymbol{\mu}^*$ driven by a normal distribution, $\boldsymbol{\mu}_0 \sim \mathcal{N}(\boldsymbol{\mu}^*, \Sigma)$, where Σ is the covariance matrix for the motion parameters. We assume that all motion parameters have the same variance σ, $\Sigma = \sigma \mathtt{I}$, where σ takes values between 0.5 and 4.0. This means that the average initial re-projection error for our experiments varies approximately between 2 and 20 pixels (see Table 4). We generate 48 000 ground truth and starting points pairs following Algorithm 6. The ground truth parameters are also used to render 48 000 images of the moving plane using `POVRAY`.

Dataset	$\Psi_{\mathtt{R}}$	$\Psi_{\mathbf{t}}$
DS1	$(0, 10)$	$(0, 10)$
DS2	$(0, 30)$	$(0, 20)$
DS3	$(30, 50)$	$(20, 30)$

Table 3 Ranges of parameters for synthetic datasets.

Algorithm 6 Creating the synthetic datasets.

```
for $\tau$ in {DS1,DS2,DS3 } do
    for $\sigma = 0.5$ to $4.0$ step $0.5$ do
        for $i = 1$ to $2\,000$ do
            Randomly sample the rotation Euler angles $\Delta^{(i)} \sim (\Psi_{\mathtt{R}}^{\tau} \cup -\Psi_{\mathtt{R}}^{\tau})$ for $\Delta = \{\alpha, \beta, \gamma\}$.
            Randomly sample the translation parameters $\Delta^{(i)} \sim (\Psi_{\mathtt{R}}^{\tau} \cup -\Psi_{\mathtt{R}}^{\tau})$ for $\Delta = \{t_x, t_y, t_z\}$.
            Store the ground truth sample $\boldsymbol{\mu}_i^* = (\alpha^{(i)}, \beta^{(i)}, \gamma^{(i)}, t_x^{(i)}, t_y^{(i)}, t_z^{(i)})^\top$.
            Generate the corresponding initial guess with Gaussian noise, $\boldsymbol{\mu}_0^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}^*, \Sigma)$.
        end for
    end for
end for
```

5.4 Plots

For each algorithm, we run the 48 000 trials with the same experimental conditions: starting point of the optimization, ground truth parameters, and textured plane image. In each experiment we record the image re-projection error at each iteration of the optimization loop and the total time used in the optimization. For each dataset, we generate four plots with these data:

- *Accuracy plot.* We plot the average final *image re-projection error* per noise value, for those experiments that successfully converged.
- *Robustness plot.* We plot the percentage of converged experiments for each noise value. This provides us with information about the *frequency of convergence* of each algorithm.
- *Efficiency plot.* We plot the average execution time against noise values for the whole optimization.
- *Rate of convergence plot.* We plot the average re-projection error for each iteration of the algorithms, for those experiments for which the optimization successfully converged. In this case, we limit the iterations to 6 to ease the visualization.

We consider that an algorithm has *successfully converged* (i.e., the algorithm has reached an acceptable solution) if the final re-projection error is below five pixels. To provide a fair comparison, we only plot the results from those experiments that successfully converged *for all the algorithms simultaneously.* This assumption eliminates non-significant extreme values that could bias the average error or timing plots. Obviously,

Table 2 Summary of experiments performed

Algorithm	Warp	Update	Constant Jacobian	CA	EBCA
LKH8	8-dof homography	Additive	No	—	—
ICH8	8-dof homography	Compositional	Yes	**YES**	**YES**
LKH6	6-dof homography	Additive	No	—	—
ICH6	6-dof homography	Compositional	Yes	**NO**	**YES**
FC3DRT	rigid motion in $\mathbb{R}^3$	Compositional	No	**YES**	—
IC3DRT	rigid motion in $\mathbb{R}^3$	Compositional	Yes	**YES**	**NO**

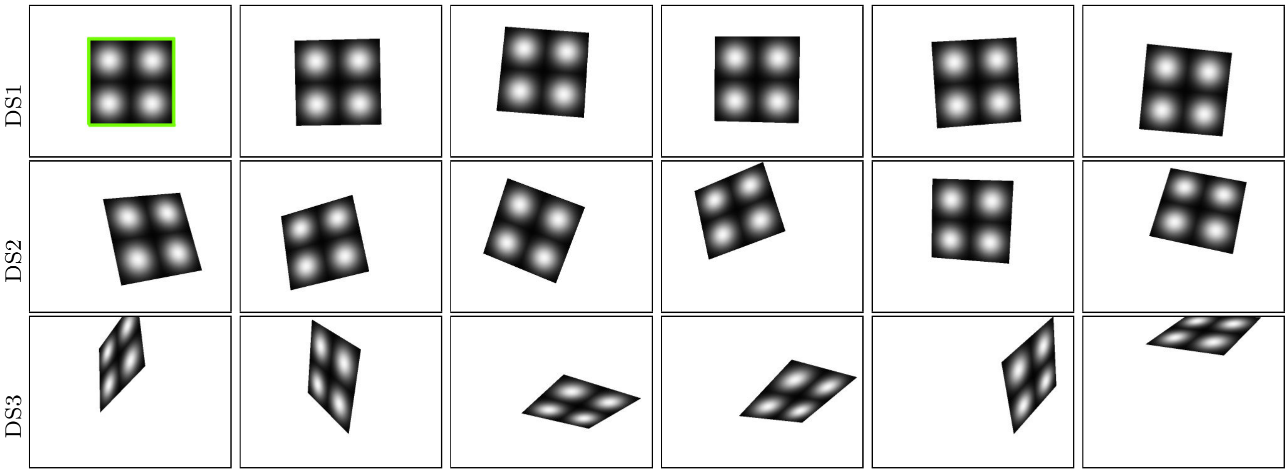

Fig. 5 Distribution of Synthetic Datasets. We select different samples from each dataset. Datasets range from DS1 (**Top**), DS2 (**Middle**), and DS3 (**Bottom**), according to Table 3. The top-left image represents the position where we compute the Jacobian for efficient methods. Observe how the successive ground truth samples increasingly depart from this location representing a wide-baseline Jacobian setting.

this assumption does not apply to the percentage of convergence plot, in which we use all data.

5.5 Results

Dataset DS1 We present the results for dataset DS1 in Fig. 6. The robustness plot for this dataset shows that for initialization noise level $\sigma < 2$, that implies an average initial error of about 10 pixels for the template (see Table 4), all algorithms converge similarly and close to 100% of the experiments. However, for higher initialization errors, ICH6, that does not satisfy the CA has the lowest percentage of convergence, 65% of the experiments for $\sigma = 4$. Efficient alignment algorithms IC3DRT and ICH8 converge slightly better, even though IC3DRT does not satisfy the EBCA. In this case, the IC approximation to the actual Jacobian is good enough for the short-baseline case. For these algorithms, when $\sigma = 4$, 80% of the tests successfully converge. Finally, non-efficient algorithms FC3DRT,LKH6,LKH8 converge close to 100% also for the highest noise levels.

The accuracy plot confirms that, for those experiments of this dataset that successfully converged, the final accuracy of all algorithms is quite similar and does not depend on the initialization.

The convergence rate of all algorithms is similar. However, the convergence time is quite different and depends on the efficiency of the algorithm. All efficient algorithms have similar convergence times. Non-efficient algorithms, FC3DRT,LKH6,LKH8, have a notably larger convergence times, being FC3DRT the slowest algorithm.

Dataset DS2 We present the results for dataset DS2 in Fig. 7. For small to moderate initialization noise levels, $\sigma < 2$, the robustness plot shows that 100% of the experiments converge for those algorithms that satisfy their requirements (LKH6, LKH8, FC3DRT and ICH8). For these algorithms, when the initialization noise is very high, $\sigma > 2$, *i.e.* an average initial error between 10 and 20 pixels, the performance of ICH8 degrades faster than that of non-efficient approaches. Finally, the convergence rate of those efficient algorithms that do not satisfy some requirement (ICH6 and IC3DRT) is notably worse than the rest, even for small initialization noise levels.

Again, the accuracy of all algorithms is similar for those registration procedures that successfully converged. However, on average worse than that of dataset DS1. The rate of convergence plot shows that all algorithms have similar convergence rates. The convergence time

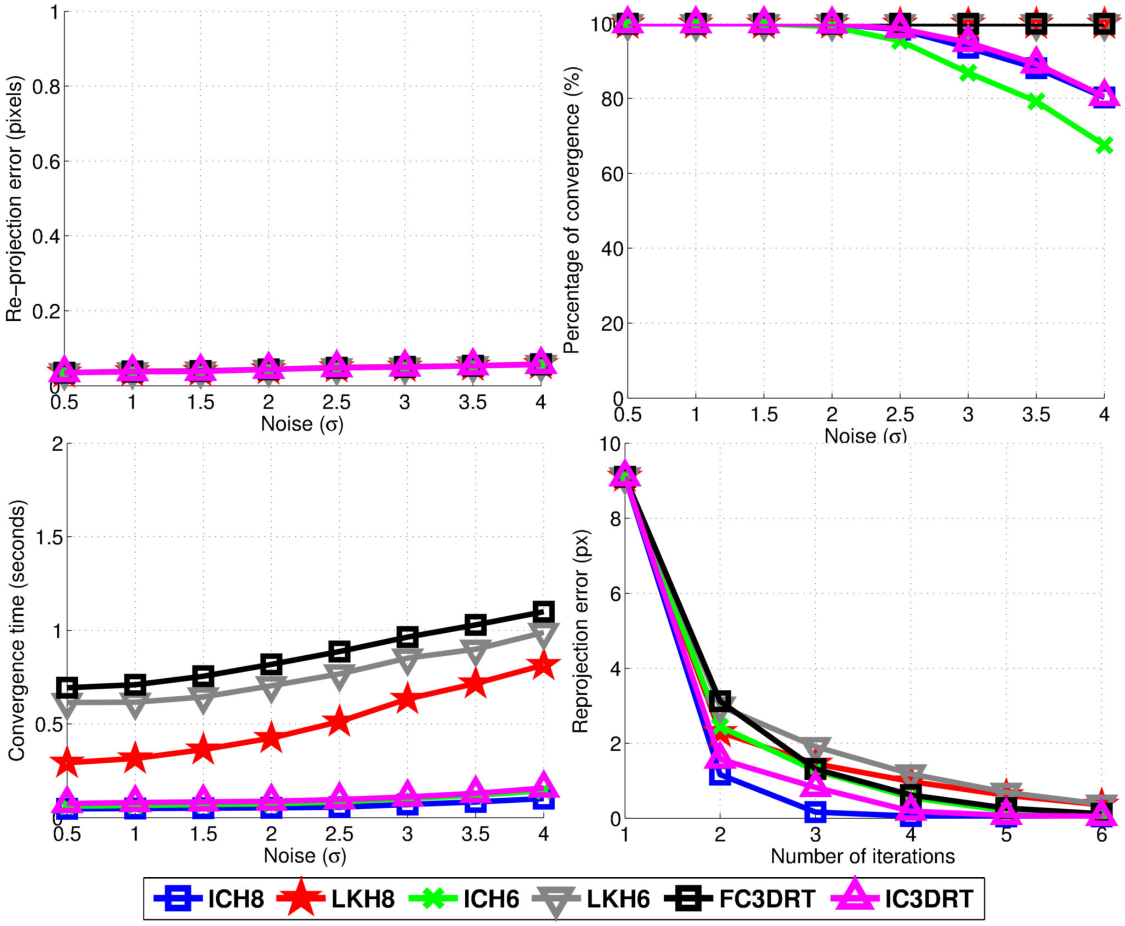

Fig. 6 Results from dataset DS1 . (Top-left) *Accuracy plot*: average re-projection error against noise standard deviation. **(Top-right)** *Robustness plot*: average frequency of convergence against noise. **(Bottom-left)** *Efficiency plot*: average convergence time against noise standard deviation.**(Bottom-right)** *Convergence plot*: Average re-projection error against number of iterations.

is again larger for non-efficient algorithms (LKH6,LKH8, and FC3DRT).

Dataset DS3 Fig. 8 shows the results for dataset DS3. Here we exclude the ICH6, and IC3DRT algorithms from the accuracy, efficiency and rate of convergence plots, since their poor convergence could bias the results.

In the robustness plot we can clearly see three groups of algorithms. The efficient algorithms that do not satisfy some requirement (IC3DRT, ICH6) do not converge for any noise level. The efficient algorithm that satisfies its requirements (ICH8) has a good convergence, above 80% of the experiments, for small to medium noise level, $\sigma < 2$. But it degrades for very high noise levels converging only 50% of the experiments for $\sigma = 4$. Non-efficient algorithms, LKH8, LKH6 and FC3DRT, converge for 98% of the experiments.

As for previous datasets, the accuracy is similar for all algorithms, but worse than that of DS1 and DS2.

The rate of convergence is also similar for all algorithms. However, the convergence time is different and depends on the efficiency of the approach. The timing results also show that all algorithms require more iterations to converge than in previous datasets. The total number of iterations is approximately 60% higher than that for dataset DS2.

6 Discussion

We first consider the impact of the satisfaction of the requirements introduced in Section 4 on the performance of efficient compositional image alignment approaches. Then we will further particularize the discussion to some typical computer vision situations such as registration, tracking and 3D surface alignment. We end this section with a discussion on the novelty of the CA and EBCA requirements.

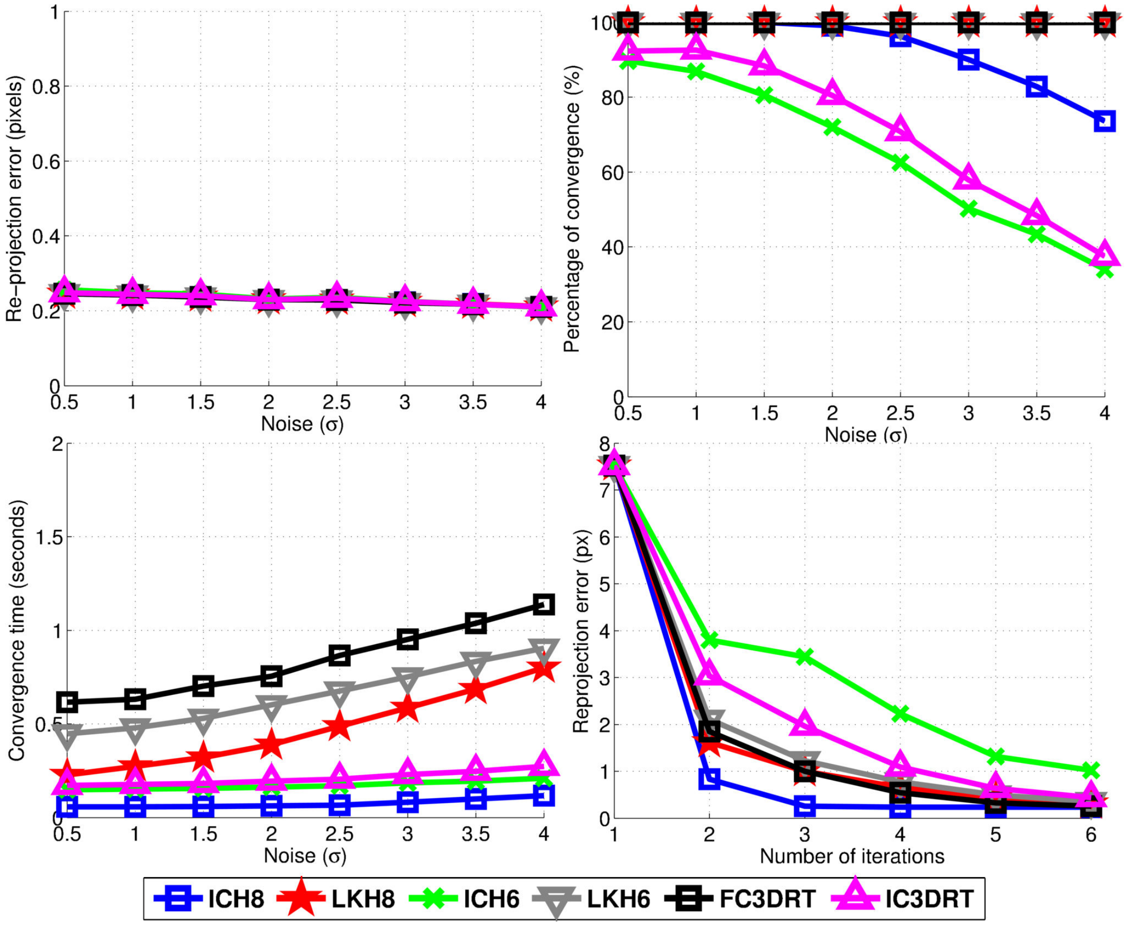

Fig. 7 Results from dataset DS2 . (Top-left) *Accuracy plot*: average re-projection error against noise standard deviation. **(Top-right)** *Robustness plot*: average frequency of convergence against noise. **(Bottom-left)** *Efficiency plot*: average convergence time against noise standard deviation.**(Bottom-right)** *Convergence plot*: Average re-projection error against number of iterations.

Table 4 Average template corner re-projection error (in pixels) vs. parameter noise for the three datasets used in the experiments.

σ	**0.50**	**1.00**	**2.00**	**3.00**	**4.00**
DS1	2.44	4.74	9.58	14.57	18.96
DS2	2.38	4.79	9.59	14.09	18.95
DS3	2.52	4.97	9.92	15.28	20.24

6.1 Impact of the satisfaction of requirements

Effects on the frequency of convergence. From our experiments we can group the algorithms into three classes based on their frequency of convergence. Non-efficient algorithms, LKH8, LKH6 and FC3DRT have the best frequency of convergence, above 95%, in all alignment settings. The efficient algorithm that satisfies its requirements, ICH8, forms the second class. It has the second best frequency of convergence, that degrades the larger the initialization noise and the wider the Jacobian baseline setting. Finally, efficient algorithms that do not satisfy their requirements, ICH6, and IC3DRT, have the worst frequency of convergence. In a wide baseline Jacobian setting they do not converge at all.

The frequency of convergence of efficient algorithms that satisfy their requirements degrades faster than their non-efficient counterparts the wider the Jacobian baseline setting. The reason for this behavior is that the proof of equivalence given in Section 4 is correct only up to a first order approximation. In addition, the computation of the Jacobian in an image far from that where the optimization takes place introduces aliasing problems that hamper the correct estimation of the Jacobian and, hence, the optimization.

When an efficient algorithm does not satisfy its requirements, the descent directions obtained with a constant Jacobian are approximately correct in a short baseline Jacobian setting, *i.e.* the setup in DS1. However, they quickly become incorrect as $\boldsymbol{\mu}_J$ drives off $\boldsymbol{\mu}$. This is the situation in a wide baseline Jacobian setting,

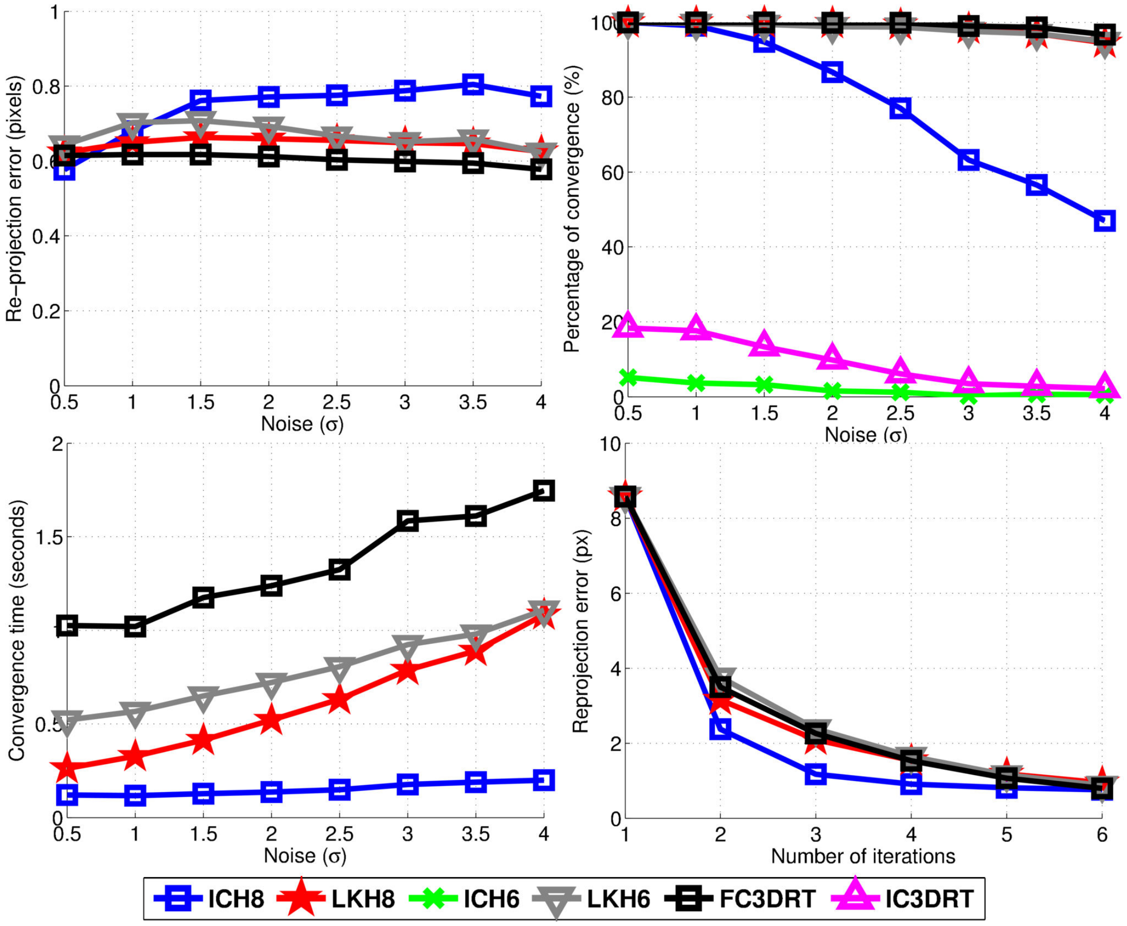

Fig. 8 Results from dataset DS3 . (Top-left) *Accuracy plot*: average re-projection error against noise standard deviation. **(Top-right)** *Robustness plot*: average frequency of convergence against noise. **(Bottom-left)** *Efficiency plot*: average convergence time against noise standard deviation.**(Bottom-right)** *Convergence plot*: Average re-projection error against number of iterations.

i.e. the setup in DS3, for which these algorithms do not converge at all, no matter how good is the initialization.

Effects on accuracy. The results show that when the optimization converges, all algorithms have similar accuracy. The final accuracy is also very similar, for all values of the initialization noise. Hence, the requirement satisfaction and initialization noise do not affect the final accuracy of the algorithm. However, it is affected by the distance between $\boldsymbol{\mu}_J$ and $\boldsymbol{\mu}^*$. In wide-baseline Jacobian settings, the final re-projection error is slightly larger than in short-baseline settings.

6.2 Efficient Registration and Tracking

Registration problems are often posed in *sort-baseline Jacobian* settings, such as those in DS1. In this case, with moderate initialization noise, the satisfaction of the CA and EBCA by the efficient compositional image alignment approach is not critical. Alignment approaches, either efficient or not, have similar convergence properties and final re-projection error. The reason for this behavior is that in a short-baseline Jacobian situation the first order approximation to the residuals obtained with the constant Jacobian is close to the real one and, hence, the descent directions computed with it are approximately correct. However, when the initialization noise level is very high, we are not in a short-baseline Jacobian situation anymore. In this case, the first order approximation is not good and efficient algorithms converge less frequently than non-efficient ones. Efficient algorithms that do not satisfy the requirements have the worst convergence. As a consequence, in typical short-baseline Jacobian image registration problems, the impact of the requirement satisfaction in the performance of an efficient alignment approach depends, basically, on the quality of the initialization.

A registration problem arising often in computer vision consist in aligning a 3D mesh model to an image, known as 2.5D alignment. In general, 2.5D alignment problems do not satisfy the EBCA. However, depending on the mesh structure and the degrees of freedom of the warping function, there are special situations in which the EBCA is satisfied, for example, aligning the mesh of a 3D plane sliding along itself (see Fig. 3(left) and Section 6.3). In a short-baseline Jacobian 2.5D alignment problem, the satisfaction of the algorithm requirements may not be critical. Hence, we can use an efficient image alignment algorithm if we have a good starting point for the optimization. Romdhani and Vetter [31] align a face 3D morphable model to a target image using an `IC` approach. Since their model does not satisfy the EBCA, they improve the convergence of the minimization by computing several Jacobians off-line and selecting the one closest to the optimization point, hence, placing their minimization in a short-baseline Jacobian setting. Similarly, Xu and Roy-Chowdhury [34] also place their alignment problem in a short-baseline setting by recomputing the optimization Jacobian.

Excluding the simplest cases, general tracking problems result in a *wide-baseline* Jacobian setting. The alignment results of DS3 prove that the satisfaction of the CA and the EBCA by an efficient alignment algorithm are critical. Hence, efficient image alignment approaches that do not satisfy their requirements cannot be used in typical tracking problems. However, efficient alignment approaches that satisfy their requirements can be safely used with low or moderate initialization noise.

6.3 On the novelty of the CA and the EBCA

In their series of works, Baker, Matthews and colleagues also study the requirements that a warping function must satisfy so it can be used within an `IC` scheme. First they require that the sets of warps contain the identity warp and be closed under inverse composition [3]. This requirement is equivalent to the CA that we introduced in Section 4.

Later, they also introduced a requirement concerning the satisfaction of the BCA on the target, $\mathcal{X}$, and on an infinitesimal neighborhood around it [4, 24]. The EBCA introduced in Section 4 is a generalization of it since it requires the satisfaction of the BCA on the target, $\mathcal{X}$, and on an infinitesimal neighborhood of the $\mathbf{g}$-warped target, $\mathcal{X}' = \mathbf{g}(\mathcal{X}, \Delta\phi)$. The EBCA is more general because it considers the case in which the warping function constrains the motion of the target to lie on the target domain, $\mathcal{V}$. In that case the gradients of the incoming image and the template can be swapped. For example, we can consider again the problem of tracking a square planar target sliding along itself and with $\mathbf{x} \in \mathcal{V} \subset \mathbb{R}^3$ (see Fig. 3(left)). This problem does not satisfy the requirement introduced by Baker, Matthews and colleagues [4, 24] since the BCA is not satisfied in an infinitesimal neighborhood outside of the plane $\mathcal{V}$. However, it does satisfy the EBCA (see Appendix B for a proof).

7 Conclusions

In this paper we have studied what makes computational efficiency afforded in `GN`-based non-linear least-squares optimization. We have introduced the concepts of equivalent and independent parameters for residual functions with two sets of parameters. We have proved that any function of residuals with equivalent and independent parameters may be efficiently optimized using a `CJGN` approach that features constant Jacobian and Hessian matrices.

We have considered the optimization problem in the context of image alignment. We proved that an alignment problem posed compositionally has equivalent parameters. It can then be solved with the `FC` algorithm. We have also introduced the EBCA, an extension of the well known BCA that enforces the equality of image values and gradients. We proved that an alignment problem that satisfies the EBCA has independent parameters. Hence, it can be efficiently solved using a `CJGN` scheme, an instance of which is the `IC` algorithm. As a consequence, `FC` and `IC` approaches are equivalent when the EBCA is satisfied.

The standard image alignment evaluation procedure, based on contaminating with noise the location of an image region and warping it back, is not adequate for studying the performance of efficient tracking algorithms, because it only contemplates the optimization problem from a local point of view. To address this issue we have introduced a new evaluation procedure, based on the the concepts of short- and wide-baseline Jacobian settings, that also considers the proximity of the template to the aligned images.

If the CA or EBCA are not satisfied, the convergence of an efficient `GN`-based alignment problem may be compromised, depending on the configuration of the images involved. In a wide-baseline Jacobian setting the optimization will diverge, no matter how close from the optimum it starts. However, in a short-baseline Jacobian setting it may converge if it has a good initialization.

Alignment approaches based on linear regression are also an instance of a `CJGN` scheme. Hence, they also require that the equivalent and independent parameters assumptions are satisfied. Otherwise, they behave like any efficient `GN`-based alignment algorithm.

Acknowledgements The authors are grateful to Pascal Fua for interesting discussions about this work. They also thank the anonymous reviewers for their comments.

A Derivative of inverse warps

Let $f(\mathbf{x}, \boldsymbol{\phi})$ be a warp function and $f^{-1}(\mathbf{x}, \boldsymbol{\phi})$ its inverse, such that $f(f^{-1}(\mathbf{x}, \boldsymbol{\phi}), \boldsymbol{\phi}) = \mathbf{x}$, where $\boldsymbol{\phi}$ is a small disturbance around the identity warp, $\boldsymbol{\phi}_0$. The derivative of this expression with respect to $\boldsymbol{\phi}$ is

$$\left.\frac{\partial f(f^{-1}(\mathbf{x}, \boldsymbol{\phi}), \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\phi}} = \mathbf{0},$$

that can be expanded using the chain rule:

$$\left.\frac{\partial f(\mathbf{x}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} + \left.\frac{\partial f(\mathbf{x}', \boldsymbol{\phi}_0)}{\partial \mathbf{x}'}\right|_{\mathbf{x}'=\mathbf{x}} \cdot \left.\frac{\partial f^{-1}(\mathbf{x}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \mathbf{0}.$$

As $f(\mathbf{x}, \boldsymbol{\phi}_0)$ is the identity warp,

$$\left.\frac{\partial f(\mathbf{x}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} + \left.\frac{\partial f^{-1}(\mathbf{x}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \mathbf{0}.$$

Finally,

$$\left.\frac{\partial f(\mathbf{x}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = -\left.\frac{\partial f^{-1}(\mathbf{x}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0}.$$

B In-plane translation

We consider the case of a plane $\boldsymbol{\pi}$ that moves perpendicular to its normal $\mathbf{n}$ at a distance d from the origin. The set of points of the plane are $\mathcal{V} = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{n}^\top \mathbf{x} + d = 0\}$, that is a two-dimensional surface embedded in $\mathbb{R}^3$, and therefore, it is a closed set. The support set is a finite subset of $\mathcal{V}$, $\mathcal{X} \subset \mathcal{V}$.

The in-plane translation has two degrees of freedom. Thus, the pair of warps $\mathbf{f}$ and $\mathbf{g}$ are parametrized respectively by $\boldsymbol{\mu} \in \mathbb{R}^3$ and $\Delta\boldsymbol{\phi} \in \mathbb{R}^2$. The two warps are $\mathbf{f}(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{x} + \boldsymbol{\mu}$ and $\mathbf{g}(\mathbf{x}, \Delta\boldsymbol{\phi}) = \mathbf{x} + [\mathbf{u}\ \mathbf{v}] \cdot \Delta\boldsymbol{\phi}$, where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$ are two independent vectors perpendicular to $\mathbf{n}$.

We will prove that this system satisfies both the CA and the EBCA. The CA states that, for any $\boldsymbol{\mu}$ and $\Delta\boldsymbol{\phi}$, there exists a $\boldsymbol{\mu}'$ such that $\mathbf{f}(\mathbf{x}, \boldsymbol{\mu}') = \mathbf{f}(\mathbf{g}(\mathbf{x}, \Delta\boldsymbol{\phi}), \boldsymbol{\mu})$. This is trivially proved taking $\boldsymbol{\mu}' = \boldsymbol{\mu} + [\mathbf{u}\ \mathbf{v}] \cdot \Delta\boldsymbol{\phi}$. The identity $\mathbf{g}$-warp is obtained for $\Delta\boldsymbol{\phi}_0 = [0\ 0]^T$.

To prove the EBCA, we write the expression for Requirement 2

$$\left.\frac{\partial I[\mathbf{x}, t]}{\partial \mathbf{x}}\right|_{\mathbf{x}=f(\mathcal{X}, \boldsymbol{\mu})} \cdot \left.\frac{\partial \mathbf{f}(\mathbf{x}, \boldsymbol{\mu})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathcal{X}} \cdot \left.\frac{\partial \mathbf{g}(\mathcal{X}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0} = \left.\frac{\partial T[\mathbf{x}]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathcal{X}} \cdot \left.\frac{\partial \mathbf{g}(\mathcal{X}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{\phi}_0}. \tag{22}$$

The derivative of the $\mathbf{f}$-warp with respect to $\mathbf{x}$ is the 3 × 3 identity. Also, the derivative of the $\mathbf{g}$-warp with respect to $\Delta\boldsymbol{\phi}$ is $[\mathbf{u}\ \mathbf{v}]$. Therefore, (22) becomes

$$\left.\frac{\partial I[\mathbf{x}, t]}{\partial \mathbf{x}}\right|_{\mathbf{x}=f(\mathcal{X}, \boldsymbol{\mu})} \cdot [\mathbf{u}\ \mathbf{v}] = \left.\frac{\partial T[\mathbf{x}]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathcal{X}} \cdot [\mathbf{u}\ \mathbf{v}]. \tag{23}$$

We do not know I nor T but, thanks to the brightness constancy assumption, we know a relation between them $I[f(\mathbf{x}, \boldsymbol{\mu}), t] = T[\mathbf{x}]$, $\forall \mathbf{x} \in \mathcal{V}$. The partial derivatives of two functions that are equal in a closed subset $\mathcal{V}$ of their domain are not, in general, equal in that subset. However, the partial derivatives projected onto $\mathcal{V}$ are equal. Thus, given a projection matrix $\boldsymbol{\Pi}$ onto the plane $\mathcal{V}$ we have that:

$$\left.\frac{\partial I[\mathbf{x}, t]}{\partial \mathbf{x}}\right|_{\mathbf{x}=f(\mathcal{X}, \boldsymbol{\mu}_t)} \cdot \boldsymbol{\Pi} = \left.\frac{\partial T[\mathbf{x}]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathcal{X}} \cdot \boldsymbol{\Pi}.$$

Since we can choose $\boldsymbol{\Pi} = [\mathbf{u}\ \mathbf{v}]$, expression (23) is true. This proves the EBCA.

References

1. Amberg, B., Vetter, T.: On compositional imge alignment, with an application to active appearance models. In: Proc. of CVPR (2009)
2. Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. In: Proc. of CVPR, vol. 1, pp. 1090–1097. IEEE (2001)
3. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifiying framework. International Journal of Computer Vision **56**(3), 221–255 (2004)
4. Baker, S., Patil, R., Cheung, G., Matthews, I.: Lucas-kanade 20 years on: Part 5. Tech. Rep. CMU-RI-TR-04-64, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2004)
5. Bartoli, A.: Groupwise geometric and photometric direct image registration. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(12), 2098 – 2108 (2008)
6. Benhimane, S., Ladikos, A., Lepetit, V., Navab, N.: Linear and quadratic subsets for template-based tracking. In: Proc. of CVPR (2007)
7. Benhimane, S., Malis, E.: Homography-based 2D visual tracking and servoing. International Jounal of Robotics Research **26**(7), 661–676 (2007)
8. Brooks, R., Arbel, T.: Generalizing inverse compositional and esm image alignment. International Journal of Computer Vision **11**(87), 191–212 (2010)
9. Buenaposada, J., Muñoz, E., Baumela, L.: Efficient illumination independent appearance-based face tracking. Image and Vision Computing **27**(5), 560–578 (2009)
10. Buenaposada, J.M., Baumela, L.: Real-time tracking and estimation of plane pose. In: Proc. of ICPR, vol. II, pp. 697–700. IEEE, Quebec, Canada (2002)
11. Buenaposada, J.M., Muñoz, E., Baumela, L.: Efficient appearance-based tracking. In: Proc. CVPR-Workshop on Nonrigid and Articulated Motion. IEEE (2004)
12. Cobzas, D., Jagersand, M., Sturm, P.: 3D SSD tracking with estimated 3d planes. Image and Vision Computing **27**(1-2), 69–79 (2009)
13. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(6), 681–685 (2001)
14. Dowson, N., Bowden, R.: Mutual information for lucas-kanade tracking (MILK): An inverse compositional formulation. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(1), 180–185 (2008)
15. Gonzalez-Mora, J., Guil, N., De la Torre, F.: Efficient image alignment using linear appearance models. In: Proc. of CVPR (2009)
16. Gross, R., Matthews, I., Baker, S.: Active appearance models with occlusion. Image and Vision Computing **24**(6), 593–604 (2006)

17. Hager, G., Belhumeur, P.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(10), 1025–1039 (1998)
18. Hinterstoisser, S., Lepetit, V., Benhimane, S., Fua, P., Navab, N.: Learning real-time perspective patch rectification. International Journal of Computer Vision **91**(1), 107–130 (2011)
19. Holzer, S., Ilic, S., Navab, N.: Multilayer adaptive linear predictors for real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(1), 105–117 (2013)
20. Holzer, S., Pollefeys, M., Ilic, S., Tan, D.J., Navab, N.: Online learning of linear predictors for real-time tracking. In: Proc. European Conference on Computer Vision. Firenze, Italy (2012)
21. Jurie, F., Dhome, M.: Hyperplane approximation for template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 996–100 (2002)
22. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. of Int. Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
23. Matthews, I., Baker, S.: Active appearance models revisited. International Journal of Computer Vision **60**(2), 135–164 (2004)
24. Matthews, I., Xiao, J., Baker, S.: 2D vs. 3D deformable face models: Representational power, construction, and real-time fitting. International Journal of Computer Vision **75**(1), 93–113 (2007)
25. Megret, R., Authesserre, J., Berthoumieu, Y.: Bidirectional composition on lie groups for gradient-based image alignment. IEEE Transactions on Image Processing **19**(9), 2369–2381 (2010)
26. Muñoz, E., Buenaposada, J.M., Baumela, L.: Efficient model-based 3D tracking of deformable objects. In: Proc. of ICCV, vol. I, pp. 877–882. Beijing, China (2005)
27. Muñoz, E., Buenaposada, J.M., Baumela, L.: A direct approach for efficiently tracking with 3D morphable models. In: Proc. of ICCV, vol. I. Kyoto, Japan (2009)
28. Navarathna, R., Sridharan, S., Lucey, S.: Fourier active appearance models. In: Proc. of ICCV (2011)
29. Nguyen, M.H., De la Torre, F.: Metric learning for image alignment. International Journal of Computer Vision **88**(1), 69–84 (2010)
30. Nocedal, J., Wright, S.: Numerical Optimization. Springer (2006)
31. Romdhani, S., Vetter, T.: Efficient, robust and accurate fitting of a 3D morphable model. In: Proc. of ICCV, vol. 1, pp. 59–66 (2003)
32. Shum, H.Y., Szeliski, R.: Construction of panoramic image mosaics with global and local alignment. International Journal of Computer Vision **36**(2), 101–130 (2000)
33. Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: Robust and efficient parametric face alignment. In: Proc. of ICCV, pp. 1847–1854 (2011)
34. Xu, Y., Roy-Chowdhury, A.K.: Inverse compositional estimation of 3D pose and lighting in dynamic scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(7), 1300 – 1307 (2008)
35. Zimmermann, K., Matas, J., Svoboda, T.: Tracking by an optimal sequence of linear predictors. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(4), 677–692 (2009)