INVITED PAPER AT THE TOPICS IN AUTOMATIC 3D MODELLING AND PROCESSING WORKSHOP, VERONA, ITALY, MARCH 2006

# Direct Image Registration With Gain and Bias

**Adrien Bartoli** ◇ Adrien.Bartoli@gmail.com
CNRS – LASMEA ◇ Clermont-Ferrand, France

## Abstract

*Image registration consists in estimating geometric and photometric transformations that align a template and an image as best as possible. The direct approach consists in minimizing the intensity discrepancy between the aligned template and image. The inverse compositional algorithm has been recently proposed for the direct estimation of groupwise geometric transformations. It is efficient in that it performs most computationally expensive calculations at the pre-computation phase.*

*We propose the gain and bias inverse compositional algorithm which estimates, along with the geometric transformation, a photometric one modeling for example global lighting change. Our algorithm preserves the efficient pre-computation-based design of the original inverse compositional one. Previous attempts at incorporating appearance variations to the inverse compositional algorithm spoils this property.*

*We report experimental results on simulated and real data, showing the improvement in computational efficiency of our algorithm compared to previous ones.*

## 1. Introduction

Image registration is the task of applying some transformations to a template and / or an image so that they match as best as possible. This can be seen as the computation of some geometric transformation, for example an homography, used to deform the image to model camera pose, and some photometric transformation, applied to the pixel intensities, for example gain and bias to model global lighting.

Image registration has been an important research topic for the past decades. It is central to many tasks in computer vision, medical imaging, augmented reality and robotics.

Broadly speaking, two approaches have been proposed: the feature-based and the direct approaches. The feature-based approach, see *e.g.* [8], relies on abstracting the input images by the geometric location of a set of carefully chosen, salient features. The direct approach, see *e.g.* [6], uses the intensity of all pixels in the region of interest.

This paper focuses on the direct approach, and brings as its main contribution a computationally efficient registration algorithm dealing with gain and bias, based on the inverse compositional principle of Baker *et al.* [2].

The geometric registration problem is the minimization of a nonlinear least squares error function, given by the discrepancy in pixel intensities, between the template $\mathcal{T}$ and the image $\mathcal{I}$, warped onto the template by the geometric transformation to be estimated. The geometric transformation, denoted $\mathcal{G}$, maps a pixel $\mathbf{q}$ in the region of interest $\mathcal{R}$ defined in the template to the corresponding pixel $\mathcal{G}(\mathbf{q};\mathbf{g})$ in the image. Vector $\mathbf{g}$ encapsulates its parameters. We expect that given an 'appropriate' parameter vector $\mathbf{g}$, $\mathcal{T}[\mathbf{q}]$ is 'close to' $\mathcal{I}[\mathcal{G}(\mathbf{q};\mathbf{g})]$, for all $\mathbf{q} \in \mathcal{R}$. The direct image registration problem is thus formally posed as:

$$\min_{\mathbf{g}} \sum_{\mathbf{q}\in\mathcal{R}} (\mathcal{T}[\mathbf{q}] - \mathcal{I}[\mathcal{G}(\mathbf{q};\mathbf{g})])^2. \quad (1)$$

Note that other error functions can be used, to deal for example with outliers, see *e.g.* [4]. Most algorithms linearize each term in the transformation parameters $\mathbf{g}$, and iteratively update an initial guess by solving linear least squares problems. The popular Lucas-Kanade algorithm [7] and work by Bergen *et al.* [3] fall into this category. Baker *et al.* [2] have recently proposed an efficient algorithm for solving problem (1), the *inverse compositional algorithm*, using a Gauss-Newton, local approximation to the error function. The efficiency stems from the fact that the Hessian matrix[1] involved in the normal equations is constant. Its inverse can thus be pre-computed.

The above-derived formulation (1) suffers from the fact that it does not take into account photometric changes, *i.e.* changes in the intensity of the pixels. These changes occur for example when the lighting changes between acquisition of the template and the image. They are modeled by a transformation $\mathcal{P}$ with parameter vector $\mathbf{p}$, and give rise to the

[1] We use the expression 'Hessian matrix' for the Gauss-Newton approximation to the true Hessian matrix.

following minimization problem:

$$\min_{\mathbf{g},\mathbf{p}} \sum_{\mathbf{q}\in\mathcal{R}} (\mathcal{P}(\mathcal{T}[\mathbf{q}];\mathbf{p}) - \mathcal{I}[\mathcal{G}(\mathbf{q};\mathbf{g})])^2. \quad (2)$$

The photometric transformation is typically chosen as an affine transformation modeling gain and bias, and accounting for global intensity changes between the template and the image:

$$\mathcal{P}(v;\mathbf{p}) = av + b \qquad \text{with} \qquad \mathbf{p}^\mathsf{T} = (a \;\; b). \quad (3)$$

The contribution of this paper is an efficient method for solving problem (2), the registration problem with gain and bias. The proposed method is dubbed the *gain and bias inverse compositional algorithm*. It is based on the inverse compositional approach of Baker *et al.* [2] and is thus applicable to the registration of images related by groupwise geometric transformations such as homographies.

Estimating gain and bias jointly with geometric registration parameters makes the Hessian matrix vary accross the iterations. Previous work thus re-estimate and invert it at each iteration: this is the *simultaneous inverse compositional algorithm* of Baker *et al.* [1], which not only deals with gain and bias but also with general linear appearance variations.

We show that the Hessian matrix has a strong block structure with blocks constant up to some scale factors, depending on the gain. From this analysis, we derive an algorithm allowing us to pre-compute a block-wise inverse of the Hessian matrix. The normal equations are then solved by simply multiplying the right hand side by some constant, appropriately rescaled matrices, which is very efficient in terms of computational cost. We underline that our algorithm performs exactly the same calculations as the simultaneous inverse compositional algorithm does. Experimental results show that the computational cost is reduced by factors of at least 2.

**Paper organization.** We introduce background material, namely the inverse compositional and the simultaneous inverse compositional algorithms of Baker *et al.* in §2. We present our gain and bias inverse compositional algorithm in §3. We report experimental results on simulated and real data in §4. A discussion is provided in §5. The parameterization of homographic warps is detailled in §A.

**Notation.** Vectors are denoted using bold fonts, *e.g.* $\mathbf{q}$, matrices using sans-serif fonts, *e.g.* $\mathsf{E}$, and scalars in italics, *e.g.* $a$. We deal with grey-level images only: the template and image, respectively denoted $\mathcal{T}$ and $\mathcal{I}$, are seen as functions from $\mathbb{R}^2$ to $\mathbb{R}$. For instance, $\mathcal{T}[\mathbf{q}]$ is the intensity at location $\mathbf{q} \in \mathbb{R}^2$. Bilinear interpolation is used for subpixel coordinates. The geometric and photometric transformations are respectively denoted $\mathcal{G}$ and $\mathcal{P}$, with respective parameter vectors $\mathbf{g}$ and $\mathbf{p}$. The geometric transformation is also called the warp.

## 2. The Inverse Compositional Algorithm

Baker *et al.* have recently published a series of five papers on direct image registration. In the first one [2], they propose the efficient *inverse compositional algorithm*. Baker *et al.* show in [1] that the efficiency is lost if appearance variations, in particular gain and bias transformations, are incorporated in the general algorithm, making it much more computationally expensive. Below, we describe this algorithm in details since it forms the basis for the one we propose.

### 2.1. Principle

The inverse compositional algorithm is an iterative procedure with, as is often the case for registration algorithms, three main steps in its inner loop:

1. **Image warping.** Warp the image on the template using the current warp parameters $\mathbf{g}$.
2. **Local registration.** Compute the local warp parameters $\boldsymbol{\delta}_g$ between the warped image and the template.
3. **Warp updating.** Update the current warp parameters by composing the current warp with the inverse of the local warp.

The main advantages of this method is that it converges rapidly, and is computationally cheap since computationally demanding calculations are pre-computed.

### 2.2. Geometric Registration

The algorithm is summarized in table 1. Let $\tilde{\mathcal{I}}$ be the warped image, *i.e.* $\tilde{\mathcal{I}}[\mathbf{q}] = \mathcal{I}[\mathcal{G}(\mathbf{q};\mathbf{g})]$. The geometric registration problem (1) is rewritten as:

$$\min_{\boldsymbol{\delta}_g} \sum_{\mathbf{q}\in\mathcal{R}} (\mathcal{T}[\mathcal{G}(\mathbf{q};\boldsymbol{\delta}_g] - \tilde{\mathcal{I}}[\mathbf{q}])^2. \quad (4)$$

Vector $\boldsymbol{\delta}_g$ represents the parameters of the local geometric transformation. The error function in problem (4) is linearized by first order Taylor expansion in $\boldsymbol{\delta}_g$ to form a Gauss-Newton approximation, giving, using the chain rule:

$$\min_{\boldsymbol{\delta}_g} \sum_{\mathbf{q}\in\mathcal{R}} (\mathcal{T}[\mathbf{q}] + \nabla\mathcal{T}[\mathbf{q}]^\mathsf{T} \left.\frac{\partial\mathcal{G}}{\partial\mathbf{g}}\right|_{\mathbf{q};\tilde{\mathbf{g}}} \boldsymbol{\delta}_g - \tilde{\mathcal{I}}[\mathbf{q}])^2,$$

where $\nabla\mathcal{T}[\mathbf{q}]$ is the $(2 \times 1)$ template gradient at $\mathbf{q}$ and $\left.\frac{\partial\mathcal{G}}{\partial\mathbf{g}}\right|_{\mathbf{q};\tilde{\mathbf{g}}}$ is the Jacobian of the warp, evaluated at $\mathbf{q}$ and at warp parameters $\tilde{\mathbf{g}}$, representing the identity warp[2]. The advantage of this formulation is that the partial derivatives of the error function are constant, and so are the gradient vectors:

$$\ell_{\mathbf{q}}^{\mathsf{T}} = \nabla\mathcal{T}[\mathbf{q}]^{\mathsf{T}} \left.\frac{\partial\mathcal{G}}{\partial\mathbf{g}}\right|_{\mathbf{q};\tilde{\mathbf{g}}}. \quad (5)$$

The normal equations induced by the linear least squares minimization problem (4) are:

$$\underbrace{\left(\sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}\ell_{\mathbf{q}}^{\mathsf{T}}\right)}_{\mathsf{E}_g} \boldsymbol{\delta}_g = \underbrace{\left(\sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}(\tilde{\mathcal{I}}[\mathbf{q}] - \mathcal{T}[\mathbf{q}])\right)}_{\mathbf{b}_g}.$$

The solution $\boldsymbol{\delta}_g = \mathsf{E}_g^{-1}\mathbf{b}_g$ for the local warp parameters can thus be computed very efficiently since the inverse of the constant Hessian matrix $\mathsf{E}_g$ can be pre-computed, as well as the gradient vectors $\ell_{\mathbf{q}}$.

Once $\boldsymbol{\delta}_g$ has been computed, the current warp parameters $\mathbf{g}$ are updated by composing the current warp with the inverse of the local warp. If one uses an homographic warp for example, then the updated parameters are given by multiplying the current homography by the inverse of the local one as detailled in §A. We write the warp update rule as:

$$\mathbf{g} \leftarrow \mathcal{U}_g(\mathbf{g}, \boldsymbol{\delta}_g).$$

The process is iterated until convergence, determined, in our experiments, by thresholding the two-norm of $\boldsymbol{\delta}_g$ by $\varepsilon = 10e - 8$, or when the update increases the error. In the latter case, the last update is cancelled before stopping the iterations.

## 2.3. Incorporating Gain and Bias

We incorporate the global, affine illumination variation model (3), referred to as gain and bias. The registration algorithm is summarized in table 2. Applying the photometric transformation to the template or to the image does not lead to exactly the same error function. They are equivalent, up to resampling issues, only when the geometric alignment is the correct one. In both cases, the Hessian matrix is not constant, thus spoiling the main advantage of the inverse compositional approach. We apply the photometric transformation to the template since it leads us to an efficient minimization algorithm in §3.

We rewrite problem (2) as:

$$\min_{\mathbf{g},\mathbf{p}} \sum_{\mathbf{q}\in\mathcal{R}} (a\mathcal{T}[\mathbf{q}] + b - \mathcal{I}[\mathcal{G}(\mathbf{q};\mathbf{g})])^2. \quad (6)$$

Using an additive update rule for the photometric parameters, *i.e.* $\mathbf{p} \leftarrow \mathbf{p} + \boldsymbol{\delta}_p$, and the inverse compositional trick

[2]The warp is generally parameterized such that $\tilde{\mathbf{g}} = \mathbf{0}$.

---

OBJECTIVE

Register an image $\mathcal{I}$ to a template $\mathcal{T}$ by computing the parameters $\mathbf{g}$ of a warp $\mathcal{G}(\mathbf{q};\mathbf{g})$ by minimizing the intensity error. Other inputs are the region of interest $\mathcal{R}$ in the template and an initial value for $\mathbf{g}$.

ALGORITHM

**Pre-computations**

1. Compute the gradient vectors $\ell_{\mathbf{q}}$ from (5) for $\mathbf{q} \in \mathcal{R}$
2. Compute the Hessian $\mathsf{E}_g = \sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}\ell_{\mathbf{q}}^{\mathsf{T}}$ and its inverse

**Iterations**

1. Warp the image $\mathcal{I}$ to $\tilde{\mathcal{I}}$ using the warp parameters $\mathbf{g}$
   - Compute the right hand side of the normal equations $\mathbf{b}_g = \sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}(\tilde{\mathcal{I}}[\mathbf{q}] - \mathcal{T}[\mathbf{q}])$
   - Solve for the update $\boldsymbol{\delta}_g = \mathsf{E}_g^{-1}\mathbf{b}_g$
2. Update the warp parameters: $\mathbf{g} \leftarrow \mathcal{U}_g(\mathbf{g}, \boldsymbol{\delta}_g)$

---

**Table 1. The inverse compositional algorithm of Baker *et al.* [2] for estimating a groupwise geometric registration.**

for the geometric parameters $\mathbf{g}$, we transform problem (6) to:

$$\min_{\boldsymbol{\delta}_g,\boldsymbol{\delta}_p} \sum_{\mathbf{q}\in\mathcal{R}} ((a+\delta_a)\mathcal{T}[\mathcal{G}(\mathbf{q};\boldsymbol{\delta}_g)] + b + \delta_b - \mathcal{I}[\mathcal{G}(\mathbf{q};\mathbf{g})])^2.$$

First order taylor expansion in $\boldsymbol{\delta}_g$ yields:

$$\min_{\boldsymbol{\delta}_g,\boldsymbol{\delta}_p} \sum_{\mathbf{q}\in\mathcal{R}} ((a+\delta_a)(\mathcal{T}[\mathbf{q}] + \ell_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\delta}_g) + b + \delta_b - \mathcal{I}[\mathcal{G}(\mathbf{q};\mathbf{g})])^2.$$

Expanding and neglecting second-order terms gives:

$$\min_{\boldsymbol{\delta}_g,\boldsymbol{\delta}_p} \sum_{\mathbf{q}\in\mathcal{R}} (a\ell_{\mathbf{q}}^{\mathsf{T}}\boldsymbol{\delta}_g + \delta_a\mathcal{T}[\mathbf{q}] + \delta_b + a\mathcal{T}[\mathbf{q}] + b - \tilde{\mathcal{I}}[\mathbf{q}])^2. \quad (7)$$

Directly solving this linear least squares problem leads to the *simultaneous inverse compositional algorithm* of Baker *et al.* [1].

Defining the complete unknown parameter update vector by $\boldsymbol{\delta}_{gp}^{\mathsf{T}} = (\boldsymbol{\delta}_g^{\mathsf{T}} \;\; \boldsymbol{\delta}_p^{\mathsf{T}})$, the normal equations are given by:

$$\underbrace{\left(\sum_{\mathbf{q}\in\mathcal{R}} \begin{pmatrix} a\ell_{\mathbf{q}} \\ \mathcal{T}[\mathbf{q}] \\ 1 \end{pmatrix} \begin{pmatrix} a\ell_{\mathbf{q}} & \mathcal{T}[\mathbf{q}] & 1 \end{pmatrix}\right)}_{\mathsf{E}_{gp}} \boldsymbol{\delta}_{gp} \quad (8)$$

$$= \underbrace{\left(\sum_{\mathbf{q}\in\mathcal{R}} \begin{pmatrix} a\ell_{\mathbf{q}} \\ \mathcal{T}[\mathbf{q}] \\ 1 \end{pmatrix} (\tilde{\mathcal{I}}[\mathbf{q}] - a\mathcal{T}[\mathbf{q}] - b)\right)}_{\mathbf{d}_{gp}}. \quad (9)$$

The Hessian matrix $\mathsf{E}_{gp}$ is clearly not constant, thus spoiling the main advantage of the inverse compositional approach. Baker *et al.* [1] propose several approximations to reduce the computational cost: the 'project out inverse compositional algorithm' and the 'normalization inverse compositional algorithm'. They show that these approximations do not perform well for high gain values, see [1].

---

OBJECTIVE

Register an image $\mathcal{I}$ to a template $\mathcal{T}$ by computing the parameters $\mathbf{g}$ of a warp $\mathcal{G}(\mathbf{q};\mathbf{g})$ and gain and bias parameters $\mathbf{p}$ by minimizing the intensity error. Other inputs are the region of interest $\mathcal{R}$ in the template and an initial value for $\mathbf{g}$ and $\mathbf{p}$.

ALGORITHM

**Pre-computations**

1. Compute the gradient vectors $\ell_{\mathbf{q}}$ from (5) for $\mathbf{q} \in \mathcal{R}$

**Iterations**

1. Warp the image $\mathcal{I}$ to $\tilde{\mathcal{I}}$ using the warp parameters $\mathbf{g}$
   - Compute and invert the Hessian matrix $\mathsf{E}_{gp}$ from (8)
   - Compute the right hand side $\mathbf{d}_{gp}$ of the normal equations from (9)
   - Solve for the update $\boldsymbol{\delta}_{gp} = \mathsf{E}_{gp}^{-1}\mathbf{d}_{gp}$
2. Update the warp parameters: $\mathbf{g} \leftarrow \mathcal{U}_g(\mathbf{g}, \boldsymbol{\delta}_g)$ and the photometric parameters: $\mathbf{p} \leftarrow \mathbf{p} + \boldsymbol{\delta}_p$

---

**Table 2. The simultaneous inverse compositional algorithm of Baker *et al.* [1] for estimating a groupwise geometric registration and gain and bias parameters. Note that the original algorithm handles general linear appearance variations.**

# 3. The Gain and Bias Inverse Compositional Algorithm

We propose an algorithm which performs exactly the same calculations as the simultaneous inverse compositional algorithm of Baker *et al.*, but which do not require one to re-compute the Hessian matrix at each iteration, thus preserving the computational advantage of the original inverse compositional algorithm. The proposed algorithm is summarized in table 3.

## 3.1. The Structure of the Hessian Matrix

We expand the Hessian matrix $\mathsf{E}_{gp}$ from equation (8):

$$\mathsf{E}_{gp} = \begin{pmatrix} a^2\mathsf{E}_g & a\mathsf{E}_c \\ a\mathsf{E}_c^\mathsf{T} & \mathsf{E}_p \end{pmatrix},$$

with $\mathsf{E}_g$, $\mathsf{E}_c$ and $\mathsf{E}_p$ depending only on the template and thus constant matrices, given by:

$$\mathsf{E}_g = \sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}\ell_{\mathbf{q}}^\mathsf{T} \qquad \mathsf{E}_c = \sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}(\mathcal{T}[\mathbf{q}] \;\; 1),$$

$$\text{and} \qquad \mathsf{E}_p = \sum_{\mathbf{q}\in\mathcal{R}} \begin{pmatrix} \mathcal{T}[\mathbf{q}]^2 & \mathcal{T}[\mathbf{q}] \\ \mathcal{T}[\mathbf{q}] & 1 \end{pmatrix}.$$

We observe that the Hessian matrix has thus a strong block structure. More precisly, all the blocks are constant up to some scale factors, depending on the gain $a$.

Similarly, the right hand side of the normal equations, defined in equation (9), is:

$$\mathbf{d}_{gp} = \begin{pmatrix} a\mathbf{d}_g \\ \mathbf{d}_p \end{pmatrix},$$

with:

$$\begin{aligned} \mathbf{d}_g &= \sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}(\tilde{\mathcal{I}}[\mathbf{q}] - a\mathcal{T}[\mathbf{q}] - b) \\ \mathbf{d}_p &= \sum_{\mathbf{q}\in\mathcal{R}} \begin{pmatrix} \mathcal{T}[\mathbf{q}] \\ 1 \end{pmatrix} (\tilde{\mathcal{I}}[\mathbf{q}] - a\mathcal{T}[\mathbf{q}] - b). \end{aligned}$$

## 3.2. Solving the Normal Equations

We propose a way to solve the normal equations allowing us to pre-compute some of the expensive steps. The solution is obtained by simple multiplication of the right hand side by rescaled constant matrices. The normal equations we want to solve are $\mathsf{E}_{gp}\boldsymbol{\delta}_{gp} = \mathbf{d}_{gp}$, or:

$$\begin{pmatrix} a^2\mathsf{E}_g & a\mathsf{E}_c \\ a\mathsf{E}_c^\mathsf{T} & \mathsf{E}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_g \\ \boldsymbol{\delta}_p \end{pmatrix} = \begin{pmatrix} a\mathbf{d}_g \\ \mathbf{d}_p \end{pmatrix}.$$

Borrowing from the standard photogrammetric block bundle adjustment technique, see *e.g.* [5], we multiply to the left by a full-rank matrix, as follows:

$$\begin{pmatrix} \mathrm{I} & 0 \\ -a\mathsf{E}_c^\mathsf{T}(a^2\mathsf{E}_g)^{-1} & \mathrm{I} \end{pmatrix} \begin{pmatrix} a^2\mathsf{E}_g & a\mathsf{E}_c \\ a\mathsf{E}_c^\mathsf{T} & \mathsf{E}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_g \\ \boldsymbol{\delta}_p \end{pmatrix}$$

$$= \begin{pmatrix} \mathrm{I} & 0 \\ -a\mathsf{E}_c^\mathsf{T}(a^2\mathsf{E}_g)^{-1} & \mathrm{I} \end{pmatrix} \begin{pmatrix} a\mathbf{d}_g \\ \mathbf{d}_p \end{pmatrix},$$

OBJECTIVE

Register an image $\mathcal{I}$ to a template $\mathcal{T}$ by computing the parameters $\mathbf{g}$ of a warp $\mathcal{G}(\mathbf{q};\mathbf{g})$ and gain and bias parameters $\mathbf{p}$ by minimizing the intensity error. Other inputs are the region of interest $\mathcal{R}$ in the template and an initial value for $\mathbf{g}$ and $\mathbf{p}$.

ALGORITHM

**Pre-computations**

1. Compute the gradient vectors $\ell_{\mathbf{q}} = \nabla\mathcal{T}[\mathbf{q}]^{\mathsf{T}} \left.\frac{\partial\mathcal{G}}{\partial\mathbf{g}}\right|_{\mathbf{q};\tilde{\mathbf{g}}}$ for $\mathbf{q}\in\mathcal{R}$
2. Compute the three blocks forming the Hessian matrix:

$$\mathsf{E}_g = \sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}\ell_{\mathbf{q}}^{\mathsf{T}} \qquad \mathsf{E}_c = \sum_{\mathbf{q}\in\mathcal{R}} \ell_{\mathbf{q}}(\mathcal{T}[\mathbf{q}]\ \ 1) \qquad \mathsf{E}_p = \sum_{\mathbf{q}\in\mathcal{R}} \begin{pmatrix} \mathcal{T}[\mathbf{q}]^2 & \mathcal{T}[\mathbf{q}] \\ \mathcal{T}[\mathbf{q}] & 1 \end{pmatrix}$$

3. Compute matrices $\mathsf{E}_p^{-1}$, $\mathsf{Z}$ and $\mathsf{Y}$:

$$\mathsf{Z} = \left(\mathsf{E}_p - \mathsf{E}_c^{\mathsf{T}}\mathsf{E}_g^{-1}\mathsf{E}_c\right)^{-1} \qquad \mathsf{Y} = -\mathsf{Z}\mathsf{E}_c^{\mathsf{T}}\mathsf{E}_g^{-1}$$

**Iterations**

1. Warp the image $\mathcal{I}$ to $\tilde{\mathcal{I}}$ using the warp parameters $\mathbf{g}$
   - Compute the right hand side of the normal equations:

$$\mathbf{d}_g = \sum_{\mathbf{q}\in\mathcal{R}} a\ell_{\mathbf{q}}(\tilde{\mathcal{I}}[\mathbf{q}] - a\mathcal{T}[\mathbf{q}] - b) \qquad \mathbf{d}_p = \sum_{\mathbf{q}\in\mathcal{R}} \begin{pmatrix} \mathcal{T}[\mathbf{q}] \\ 1 \end{pmatrix} (\tilde{\mathcal{I}}[\mathbf{q}] - a\mathcal{T}[\mathbf{q}] - b)$$

   - Solve for the update:

$$\boldsymbol{\delta}_p = \mathsf{Z}\mathbf{d}_p + \mathsf{Y}\mathbf{d}_g \qquad \boldsymbol{\delta}_g = \frac{1}{a}\mathsf{E}_p^{-1}(\mathbf{d}_g - \mathsf{E}_c\boldsymbol{\delta}_p)$$

2. Update the warp and photometric parameters:

$$\mathbf{g} \leftarrow \mathcal{U}_g(\mathbf{g}, \boldsymbol{\delta}_g) \qquad \mathbf{p} \leftarrow \mathbf{p} + \boldsymbol{\delta}_p$$

**Table 3. The proposed gain and bias inverse compositional algorithm for estimating a groupwise geometric registration and gain and bias parameters.**

giving:

$$\begin{pmatrix} a^2\mathsf{E}_g & a\mathsf{E}_c \\ 0 & \mathsf{E}_p - a\mathsf{E}_c^\mathsf{T}(a^2\mathsf{E}_g)^{-1}a\mathsf{E}_c \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_g \\ \boldsymbol{\delta}_p \end{pmatrix} = \begin{pmatrix} a\mathbf{d}_g \\ \mathbf{d}_p - a^2\mathsf{E}_c^\mathsf{T}(a^2\mathsf{E}_g)^{-1}\mathbf{d}_g \end{pmatrix}.$$

This equation simplifies to:

$$\begin{pmatrix} a^2\mathsf{E}_g & a\mathsf{E}_c \\ 0 & \mathsf{E}_p - \mathsf{E}_c^\mathsf{T}\mathsf{E}_g^{-1}\mathsf{E}_c \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_g \\ \boldsymbol{\delta}_p \end{pmatrix} = \begin{pmatrix} a\mathbf{d}_g \\ \mathbf{d}_p - \mathsf{E}_c^\mathsf{T}\mathsf{E}_g^{-1}\mathbf{d}_g \end{pmatrix}.$$

The solution for the photometric parameters $\boldsymbol{\delta}_p$ is obtained directly from the second set of equations as:

$$\boldsymbol{\delta}_p = \mathsf{Z}\mathbf{d}_p + \mathsf{Y}\mathbf{d}_g,$$

where $\mathsf{Z}$ and $\mathsf{Y}$ are constant matrices, given by:

$$\begin{aligned} \mathsf{Z} &= \left(\mathsf{E}_p - \mathsf{E}_c^\mathsf{T}\mathsf{E}_g^{-1}\mathsf{E}_c\right)^{-1} \\ \mathsf{Y} &= -\mathsf{Z}\mathsf{E}_c^\mathsf{T}\mathsf{E}_g^{-1}. \end{aligned}$$

The solution for the geometric parameters $\boldsymbol{\delta}_g$ is given, from the first set of equations, by:

$$\begin{aligned} a^2\mathsf{E}_g\boldsymbol{\delta}_g &= a\mathbf{d}_g - a\mathsf{E}_c\boldsymbol{\delta}_p \\ \boldsymbol{\delta}_g &= \frac{1}{a}\mathsf{E}_p^{-1}(\mathbf{d}_g - \mathsf{E}_c\boldsymbol{\delta}_p). \end{aligned}$$

In this equation, matrix $\mathsf{E}_p^{-1}$ is constant and can be precomputed.

# 4. Experimental Results

We compared the simultaneous inverse compositional algorithm of Baker *et al.* and the gain and bias inverse compositional algorithm we propose, as described in tables 2 and 3 respectively, in the case of homographic warps, see §A. Note that both algorithms produce *exactly* the same results but with different computation times. Our experiments are designed to assess to which extent these differences are significant. We refer the reader to [1, 2] for a thorough set of experiments on the behaviour of a great variety of different algorithms. The cost of an iteration is constant for each algorithm. We used our own, fairly optimized implementation in MATLAB.

## 4.1. Simulated Data

Figure 1 shows the computational time of an iteration when varying image side length. The ranges of image side lengths are 10 to 100 and 100 to 1000 respectively for the left and right hand side graphs. We observed that their is a factor of at least 2 between the two algorithms, in favor of the gain and bias inverse compositional algorithm, in all cases.

## 4.2. Real Data

We compared the algorithms on several sets of images. We one of them, we show results. The template and the image are both $600 \times 800$. They are shown on figure 3, together with the region of interest. The region of interest contains 255,210 pixels. Figure 2 shows the error in intensity through the 28 iterations that were necessary to register the images. Figure 4 shows the error image at different iterations. The photometric parameters that were computed are $a = 0.98$ and $b = 3.77$, and the final RMS intensity error is 4.13. The computational time needed by the simultaneous inverse compositional algorithm was 76.91 seconds, while the gain and bias inverse compositional algorithm took 38.40 seconds.

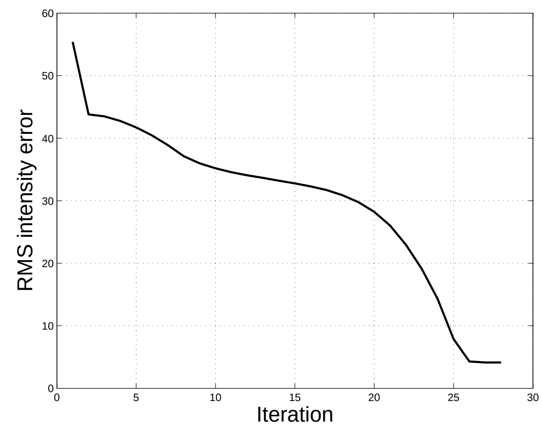



**Figure 2. Error in intensity through the iterations for the images shown in figure 3.**

# 5. Discussion

The proposed algorithm efficiently extends the inverse compositional algorithm of Baker *et al.* [2] to handle gain and bias. The computational time is reduced by a factor of at least 2 compared to the general linear appearance variations algorithm of Baker *et al.* [1].

There are several important issues that need to be investigated. The first one is about numerical conditioning: the elements of the Hessian matrices have different orders of magnitude, from 1 to $\mathcal{O}(k^2 s)$, where $k$ is the image side length (in pixels) and $s$ the maximum image intensity (in practice we expect $s$ to be close to 255). Similarly to the normalization used to improve the conditioning in the eight point algorithm [5], we naturally wonder if normalizing the image coordinates and intensity can improve the numerical

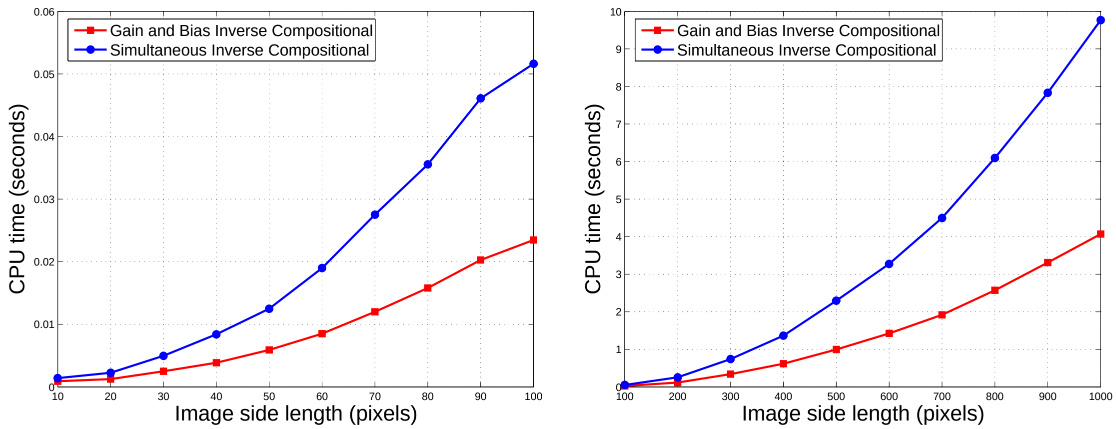
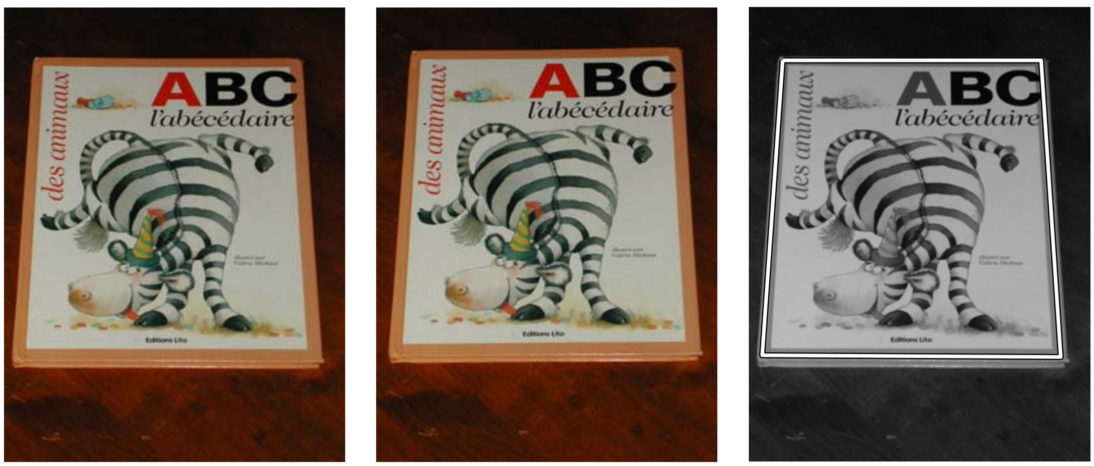



**Figure 1. Computational time of an iteration versus image side length.**




Template | Image | Region of interest

**Figure 3. Real images used in the experiments.**

Iteration 1
RMS error: 55.41

Iteration 10
RMS error: 34.56

Iteration 19
RMS error: 28.22

Iteration 28
RMS error: 4.13

**Figure 4. The error image at different iterations and corresponding RMS error on the intensity.**

stability and thus the convergence properties of the algorithms.

The second issue is about using color images, *i.e.* determine if the proposed algorithm extends to deal with individual gain and bias for each color channel, or even for full linear combinations of the color channels.

The MATLAB code used to produce the experimental results in this paper is available for download on the web homepage of the author.

## References


[1] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, Robotics Institute, Carnegie Mellon University, November 2003.

[2] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, February 2004.

[3] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, 1992.

[4] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

[6] M. Irani and P. Anandan. About direct methods. In *Workshop on Vision Algorithms: Theory and Practice*, 1999.

[7] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981.

[8] P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *Workshop on Vision Algorithms: Theory and Practice*, 1999.


## A. Parameterizing Homographic Warps

The homographic warp, denoted $\mathcal{H}$, has 9 parameters defined up to scale. In homogenous coordinates, it is represented by a $(3 \times 3)$ homography matrix $\mathsf{H}$. The representation of the warp by an homography matrix makes it easy to invert a warp or compose two warps, as required by the inverse compositional algorithm, respectively by inverting the homography matrix and by multiplying the two homography matrices.

Following [2], the local homography matrix is parameterized by an 8-vector $\boldsymbol{\delta}_h$ as:

$$\Delta\mathsf{H} \sim \mathrm{I} + \begin{pmatrix} \delta_{h,1} & \delta_{h,2} & \delta_{h,3} \\ \delta_{h,4} & \delta_{h,5} & \delta_{h,6} \\ \delta_{h,7} & \delta_{h,8} & 0 \end{pmatrix}. \tag{10}$$

The corresponding warp is:

$$\mathcal{H}(\mathbf{q}; \boldsymbol{\delta}_h) = \frac{1}{\delta_{h,7}q_1 + \delta_{h,8}q_2} \begin{pmatrix} (1+\delta_{h,1})q_1 + \delta_{h,2}q_2 + \delta_{h,3} \\ \delta_{h,4}q_1 + (1+\delta_{h,5})q_2 + \delta_{h,6} \end{pmatrix}.$$

Note that $\boldsymbol{\delta}_h = \mathbf{0}$ corresponds to the identity warp since $\mathcal{H}(\mathbf{q}; \mathbf{0}) = \mathbf{q}$.

In practice, we represent the warp by the $(3 \times 3)$ homography matrix $\mathsf{H}$, and implement the update rule as:

$$\mathcal{U}_g(\mathsf{H}, \boldsymbol{\delta}_h) = \mathsf{H} \cdot \Delta\mathsf{H},$$

where $\Delta\mathsf{H}$ is given by equation (10).

The Jacobian of the warp, evaluated around the identity warp, is given by:

$$\left.\frac{\partial \mathcal{H}}{\partial \boldsymbol{\delta}_h}\right|_{\mathbf{q};\mathbf{0}} = \begin{pmatrix} q_1 & q_2 & 1 & 0 & 0 & 0 & -q_1^2 & -q_1 q_2 \\ 0 & 0 & 0 & q_1 & q_2 & 1 & -q_1 q_2 & -q_2^2 \end{pmatrix}.$$