Dense Methods for Image Alignment with an Application to 3D Tracking

EPFL Report 197866

Alberto Crivellaro[1], Pascal Fua[1], and Vincent Lepetit[2]

[1]Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland
[2]Institute for Computer Graphics and Vision, Graz University of Technology

March 25, 2014

Abstract

This survey focuses on a class of methods for image alignment based on a global, iterative optimization. We update the important survey from Baker and Matthews [2], synthetically describing the existing methods (included the recent Efficient Second Order Method, not covered by [2]), comparing their theoretical aspects, computational costs and implementation issues, and describing an example of application to 3D tracking, involving a complex, non-linear warp.

1 Introduction

Recently, methods for image alignment based on global optimization problems have undergone a regain of interest for their accuracy and robustness, thanks to the growing power of modern computing devices, with applications such as image stitching, pose estimation [7], optical flow, object tracking, face coding, and others.

Since the seminal work of Lucas and Kanade [6], several optimization methods have been proposed, offering solutions more and more efficient and accurate, and an excellent survey from Baker and Matthews [2] already reviews them. This present survey is meant to be an update of [2]:

- we provide a synthetic description of the existing methods, highlighting both theoretical aspects and implementation issues;
- we include the Efficient Second-order Method [3], which was absent from [2], in the same framework;
- we also give an example of application to 3D tracking, involving a complex, non-linear warp.

2 Dense image alignment

Let T, I be 2 images, seen here as functions returning the value of the luminous intensity for a given pixel location $\mathbf{x}$:

$$T, I : \mathbb{R}^2 \to \mathbb{R}; \quad \mathbf{x} \mapsto T(\mathbf{x}), I(\mathbf{x}) \,. \tag{1}$$

T will denote a "reference image", or "template"; I will denote an input image.

Let $\mathcal{F}$ be a family of warps, parametrized by n parameters stored in a vector $\mathbf{p}$:

$$\mathbf{W} : \mathbb{R}^2 \times \mathbb{R}^n \to \mathbb{R}^2, \quad (\mathbf{x}, \mathbf{p}) \mapsto \mathbf{W}(\mathbf{x}, \mathbf{p}) \,. \tag{2}$$

Furthermore, we introduce an explicit notation for the warped image $I_{\mathbf{p}}$:

$$I_{\mathbf{p}}(\mathbf{x}) = I(\mathbf{W}(\mathbf{x}, \mathbf{p})) \quad \forall \mathbf{x} \in \mathcal{D} \,, \tag{3}$$

where $\mathcal{D} \subset \mathbb{R}^2$ the domain of the reference image T.

As shown in Figure 1, image alignment consists in estimating the parameters $\mathbf{p}_I$ of the warp mapping the reference image T (*template*) over an input image I, such that $T(\mathbf{x}) = I_{\mathbf{p}_I}(\mathbf{x}) = I(\mathbf{W}(\mathbf{x}, \mathbf{p}_I))$.

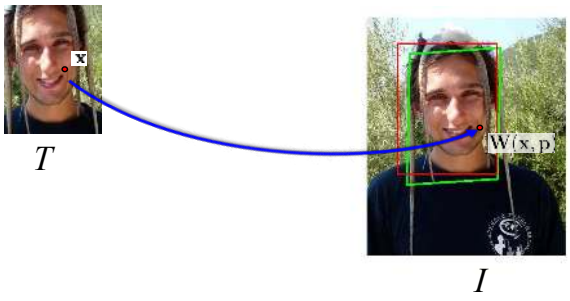


Figure 1: The image alignment problem.

This problem can be modelled through the following optimization problem:

$$\mathbf{p}_I = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}} \Big(I(\mathbf{W}(\mathbf{x}, \mathbf{p})) - T(\mathbf{x}) \Big)^2, \tag{4}$$

where the sum is extended over all, or a dense subset of, the pixels of the template. Even for very simple warps, the optimization problem (4) is highly non-linear because of the presence of the functions $T(\cdot), I(\cdot)$, and it is usually solved by iterative methods. On the other hand, this approach does not need to detect nor match any local image features, which makes dense methods particularly suited for aligning images with low textures or repetitive patterns. Examples of frequently employed families of warps are 2D and 3D translations, in-plane and off-plane rotations, affine warps, and homographies.

3 Optimization framework

All the iterative algorithms presented in this survey share the same basic structure. At iteration c, given the current estimate of the parameters $\mathbf{p}_c$, they:

1. seek for an increment $\delta\mathbf{p}$ of the current parameters, that minimizes a non-linear objective function $F(\delta\mathbf{p})$ somehow related to Equation (4);

2. approximate the non-linear objective function using a finite order development, and obtain a closed-form formula for $\delta\mathbf{p}$.

3. update the current estimate of the parameters using the computed value of $\delta\mathbf{p}$.

The above steps are iterated until some stopping criterion is met, such as $||\delta\mathbf{p}|| < \epsilon_{tol}$ for a threshold ϵ_{tol} selected by the user. The methods differ for the kind of objective function employed (*additive* or *compositional*), for the direction of the warping (*forward* or *inverse*), and for the kind of approximation employed (a second order development for the Efficient Second-order Method, a first-order one for the other methods). In Appendix D a thorough comparison of all the methods described in this survey is provided.

Depending on the different choices, the methods can be applied to different classes of warps, and have different computational costs and convergence properties. This is discussed in the next sections.

4 First order methods

4.1 Forward Additive algorithm

The most widespread dense image alignment method is the well-known Lucas-Kanade algorithm proposed in [6]. At iteration c, given a current estimate of the parameters $\mathbf{p}_c$, we seek for an increment of the parameters $\delta\mathbf{p}$ that minimizes an approximation of the objective function:

$$F_{FA}(\delta\mathbf{p}) = \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c + \delta\mathbf{p})) - T(\mathbf{x}))^2. \tag{5}$$

According to the classification proposed in [2], this is a *forward* method, because it seeks for an update of the warp of the image, and it is *additive*, because the update is summed to the current estimate of the parameters. The outline of this Forward Additive algorithm is schematically represented in Figure 2.

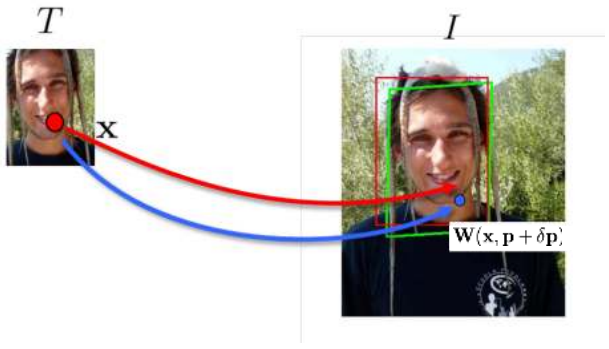


Figure 2: The Forward Additive algorithm. The current warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ for a pixel $\mathbf{x}$ is represented as a red arrow, the updated warp as a blue arrow.

Approximate solution: Equation (5) is a non-linear function with respect to $\delta\mathbf{p}$, so we approximate it by computing a first-order Taylor expansion of $I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c + \delta\mathbf{p}))$ with respect to the second argument of $\mathbf{W}$ around $\mathbf{p}_c$:

$$F_{FA}(\delta\mathbf{p}) \approx \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) + \mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)\delta\mathbf{p} - T(\mathbf{x}))^2 ; \tag{6}$$

$\mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)$ is the $n \times 1$ Jacobian matrix of F_{FA}:

$$\mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c) = \nabla I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} , \tag{7}$$

where ∇I is the gradient of I; the second term of Equation (6) is a quadratic form with respect to $\delta\mathbf{p}$; by setting its derivative equal to zero, it is possible to obtain the following closed-form solution for $\delta\mathbf{p}$:

$$\delta\mathbf{p} = H(\mathbf{p}_c)^{-1} \sum_{\mathbf{x}} \mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)^T (T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) , \tag{8}$$

where $H(\mathbf{p}_c) = \sum_{\mathbf{x}} \mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)^T \mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)$.

Parameters update: After computing an increment $\delta\mathbf{p}$ using Equation (8), the current estimate of the parameters is updated with the following additive rule:

$$\mathbf{p}_{c+1} = \mathbf{p}_c + \delta\mathbf{p} . \tag{9}$$

Assumption on the set of warps: The only requirement for the warps $\mathbf{W}(\mathbf{x}, \mathbf{p}) \in \mathcal{F}$ is to be differentiable with respect to $\mathbf{p}$.

Computational complexity: The main steps of the FA algorithms are summarized in Algorithm 4.1. The computational complexity of each step is reported as a function of the number of parameters n and the number of pixels of the template N.

Algorithm 1 Forward Additional algorithm

while $\|\|\delta\mathbf{p}\|\| > \epsilon$ **do**	
Compute $\mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)$ for all $\mathbf{x} \in \mathcal{D}$ with Equation (7)	$\mathcal{O}(nN)$
Compute $H(\mathbf{p}_c) = \sum_{\mathbf{x}} \mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)^T \mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c)$	$\mathcal{O}(n^2N)$
Compute $\delta\mathbf{p}$ with Equation (8)	$\mathcal{O}(nN + n^3)$
Update the parameters using Equation (9)	$\mathcal{O}(n)$
end while	

One iteration of the FA algorithm has a computational complexity of:

$$\mathcal{O}(n^2N + n^3) \tag{10}$$

Note that, for usual applications, $n \leq 10$, and $N \in [10^3, 10^5]$.

4.2 Forward Compositional algorithm

An alternative algorithm is the Forward Compositional algorithm (FC), proposed in [8]. At iteration c, given the current estimate of the parameters $\mathbf{p}_c$, we seek for an increment of the parameters $\delta\mathbf{p}$ minimizing

$$F_{FC}(\delta\mathbf{p}) = \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c)) - T(\mathbf{x}))^2 \,; \tag{11}$$

employing a *compositional* approach instead of the additional approach of Equation (5) can lead to better computational performances than the FA algorithm, while, under some assumptions, the convergence properties of the two methods are equivalent. This will be discussed in Section 8. The update of the warp provided by the FC algorithm is represented in Figure 3

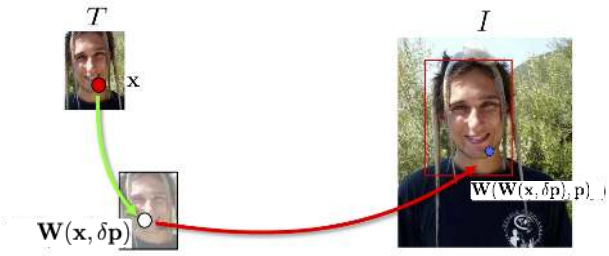


Figure 3: Updated warp in the Forward Compositional algorithm. The warp increment $\mathbf{W}(\mathbf{x}, \delta\mathbf{p})$ is shown as a green arrow.

Approximate solution: Assuming that $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$, a first-order Taylor expansion of $I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{p}), \mathbf{p}_c))$ around $\mathbf{p} = \mathbf{0}$ yields:

$$F_{FC}(\delta\mathbf{p}) \;\approx\; \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) + \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)\delta\mathbf{p} - T(\mathbf{x}))^2 \,, \tag{12}$$

where

$$\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c) = \nabla I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \frac{\partial \mathbf{W}(\mathbf{y}, \mathbf{p}_c)}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{x}} \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{0}} \,, \tag{13}$$

where the last term on the right does not depends on $\mathbf{p}_c$ and can be pre-computed.

As done in Section 4.1 for the FA algorithm, by deriving the quadratic form of Equation (12) with respect to $\delta\mathbf{p}$ and setting the derivative equal to zero yields:

$$\delta\mathbf{p} = H(\mathbf{p}_c)^{-1} \sum_{\mathbf{x}} \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)^T (T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c))) \,, \tag{14}$$

where $H(\mathbf{p}_c) = \sum_{\mathbf{x}} \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)^T \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)$.

Parameters update: Once an approximate solution $\delta\mathbf{p}$ for the minimization problem (11) has been found with Equation (14), the *warp* is updated with the compositional rule:

$$\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1}) = \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c). \tag{15}$$

If an explicit expression for $\mathbf{p}_{c+1}$ from Equation (15) cannot be found, then it is possible to estimate it, for instance, by computing $\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1})$ for a subset of the pixels and then fitting a regression model to the correspondences $\{\mathbf{x} \leftrightarrow \mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1})\}$.

Assumptions on the set of warps: In order to compute the updated warp at each iteration, the composition of 2 admissible warps must be an admissible warp, that is, $\mathcal{F}$ must be closed with respect to composition. Moreover, in order to compute $\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)$, all warps $\mathbf{W} \in \mathcal{F}$ must be differentiable. Finally, we require that the identity is an admissible warp, so that $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$ (after a re-parametrization if needed, as for the warp described in Section 7.2). That is, $\mathcal{F}$ should form a semi-group of differentiable warps.

Computational complexity: The main steps of the FC algorithms are resumed in Algorithm 2, along with the computational complexity of each step.

Algorithm 2 Forward Compositional algorithm

Pre-compute $\frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}}\Big\|_{\mathbf{p}=\mathbf{0}}$, for all $\mathbf{x} \in \mathcal{D}$	$\mathcal{O}(nN)$
while $\|\|\delta\mathbf{p}\|\| > \epsilon$ **do**	
Compute $\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)$ for all $\mathbf{x} \in \mathcal{D}$ with Equation (13)	$\mathcal{O}(nN)$
Compute $H(\mathbf{p}_c) = \sum_{\mathbf{x}} \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)^T \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)$	$\mathcal{O}(n^2N)$
Compute $H(\mathbf{p}_c)^{-1}$	$\mathcal{O}(n^3)$
Compute $\delta\mathbf{p}$ with Equation (14)	$\mathcal{O}(nN + n^2)$
Update the parameters using Equation (9)	$\mathcal{O}(n^2)$
end while	

Despite the fact that some quantities can be pre-computed, one iteration of the FC algorithm has the same computational complexity as the FA algorithm:

$$\mathcal{O}(n^2N + n^3) \tag{16}$$

Actually, the complexity of the update of the warp depends on the family of warps, in Algorithm 2 the computational complexity for affine warps ($\mathcal{O}(n^2)$) is reported; for other kinds of warps the computational cost of this step can change, but it usually it does not affect the global complexity estimation for one iteration of the algorithm.

4.3 Inverse Compositional algorithm

A major drawback of the forward algorithms is their heavy computational cost, since the matrix H has to be re-computed at each iteration. A more efficient iterative method, the Inverse Compositional algorithm (IC), has been proposed in [1]. Given the current estimate of the parameters $\mathbf{p}_c$, the IC algorithm seeks an increment $\delta\mathbf{p}$ that minimizes

$$F_{IC}(\delta\mathbf{p}) = \sum_{\mathbf{x}} (T(\mathbf{W}(\mathbf{x}, \delta\mathbf{p})) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)))^2 \,; \tag{17}$$

note that this objective function is very similar to that of the FC algorithm (11), but the roles of the image and the template are switched. Rather than seeking for an incremental warp

that makes the warped image more similar to the template, the template is warped to make it more similar to the current warped image. This trick allows to decrease the computational cost. The update of the warp in the IC algorithm is shown in Figure 4

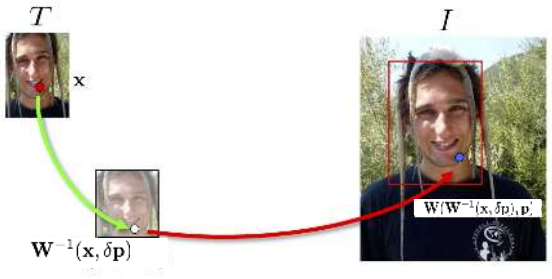


Figure 4: Updated warp provided by the Inverse Compositional algorithm. The inverse of the warp increment $\mathbf{W}(\mathbf{x}, \delta\mathbf{p})$ is shown as a green arrow.

Approximate solution: As in the previous sections, to find an approximated objective function, we perform a first order expansion of $T(\mathbf{W}(\mathbf{x}, \mathbf{p}))$ around $\mathbf{p} = \mathbf{0}$. Assuming that $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$ and setting the derivative of the resulting quadratic form equal to zero yields:

$$\delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \mathbf{J}_{IC}(\mathbf{x})^T [I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) - T(\mathbf{x})] \,, \tag{18}$$

where $H = \sum_{\mathbf{x}} \mathbf{J}_{IC}(\mathbf{x})^T \mathbf{J}_{IC}(\mathbf{x})$, and:

$$\mathbf{J}_{IC}(\mathbf{x}) = \nabla T(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} \Big|_{\mathbf{p}=\mathbf{0}} \,. \tag{19}$$

Note that $\mathbf{J}_{IC}(\mathbf{x})$ and H do NOT depend on the current parameters estimate and can be pre-computed once and for all.

Parameters update: After computing $\delta\mathbf{p}$ with Equation (18), the current warp is updated so that:

$$\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1}) = \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p})^{-1}, \mathbf{p}_c) \,. \tag{20}$$

As for the FC algorithm, if an expression for $\mathbf{p}_{c+1}$ can not be explicitly computed from Equation (15), it is possible to compute it by fitting a regression model to the correspondences $\{\mathbf{x} \leftrightarrow \mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1})\}$.

Assumptions on the set of warps: As for FC, we assume that the warps $\mathbf{W} \in \mathcal{F}$ are differentiable, that $\mathcal{F}$ is closed with respect to the composition and that the identity is an admissible warp. Moreover, the warps should be invertible and $\mathcal{F}$ should be closed under the inversion. That is, $\mathcal{F}$ must form a group.

Algorithm 3 Inverse Compositional algorithm

Pre-compute $\mathbf{J}_{IC}(\mathbf{x})$ with Equation (19) for all $\mathbf{x} \in \mathcal{D}$	$\mathcal{O}(nN)$
Pre-compute $H = \sum_{\mathbf{x}} \mathbf{J}_{IC}(\mathbf{x})^T \mathbf{J}_{IC}(\mathbf{x})$	$\mathcal{O}(n^2 N)$
Pre-compute H^{-1}	$\mathcal{O}(n^3)$
while $\|\|\delta\mathbf{p}\|\| > \epsilon$ **do**	
Compute $\delta\mathbf{p}$ with Equation (18)	$\mathcal{O}(nN + n^2)$
Update the parameters using Equation (20)	$\mathcal{O}(n^2)$
end while	

Computational complexity: The main steps of the IC algorithms and their computational complexity are reported in Algorithm 3. As for the FC algorithm, the cost of the update of the warp depends on the family of warps, but generally it does not affect the global complexity estimation. Thanks to the fact that H^{-1} and $\mathbf{J}_{IC}(\mathbf{x})$ can be pre-computed, the computational complexity for one iteration of the IC algorithm is lower than that of the forward algorithms:

$$\mathcal{O}(nN + n^2). \tag{21}$$

Notice that pre-computing H^{-1} is numerically less stable than solving a linear system involving H at each iteration, but this does not represent a problem in most part of practical applications.

4.4 Inverse Additive algorithm

An Inverse Additive algorithm (IA) has been proposed in [5], which has as a low computational complexity as the IC algorithm while employing an additive update of the parameters. Unfortunately, as we will see, this algorithm can only be applied to a very restricted set of warps. Given the current estimate of the parameters $\mathbf{p}_c$, the IA algorithm finds an increment $\delta\mathbf{p}$ that approximatively minimizes the same objective function as the Forward Additive algorithm:

$$F_{IA}(\delta\mathbf{p}) = F_{FA}(\delta\mathbf{p}) = \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c + \delta\mathbf{p})) - T(\mathbf{x}))^2 \,. \tag{22}$$

Approximate solution: We start from the first order expansion of $I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c + \delta\mathbf{p}))$ around $\mathbf{p}_c$ of Equation (6):

$$F_{IA}(\delta\mathbf{p}) \;\approx\; \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) + \mathbf{J}_{IA}(\mathbf{x}, \mathbf{p}_c)\delta\mathbf{p} - T(\mathbf{x}))^2 \,, \tag{23}$$

where:

$$\mathbf{J}_{IA}(\mathbf{x}, \mathbf{p}_c) = \mathbf{J}_{FA}(\mathbf{x}, \mathbf{p}_c) = \nabla I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} \,. \tag{24}$$

Assuming that the current parameters estimate is approximately correct, that is, $I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \approx T(\mathbf{x})$, the following approximation holds:

$$\frac{\partial I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c))}{\partial \mathbf{x}} = \nabla I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p}_c)}{\partial \mathbf{x}} \;\approx\; \nabla T \,. \tag{25}$$

Injecting this approximation in Equation (24), yields:

$$\mathbf{J}_{IA}(\mathbf{x},\mathbf{p}_c) = \nabla T(\mathbf{x}) \left(\frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p}_c)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}} \Big|_{\mathbf{p}=\mathbf{p}_c}. \tag{26}$$

Minimizing the quadratic form (23), we find the closed-form solution for $\delta\mathbf{p}$:

$$\delta\mathbf{p} = H(\mathbf{p}_c)^{-1} \sum_{\mathbf{x}} \mathbf{J}_{IA}(\mathbf{x},\mathbf{p}_c)^T (I(\mathbf{W}(\mathbf{x},\mathbf{p}_c)) - T(\mathbf{x})), \tag{27}$$

where $H(\mathbf{p}_c) = \sum_{\mathbf{x}} \mathbf{J}_{IA}(\mathbf{x},\mathbf{p}_c)^T \mathbf{J}_{IA}(\mathbf{x},\mathbf{p}_c)$. For keeping notations coherent, in the above formula we replaced $(I(\mathbf{W}(\mathbf{x},\mathbf{p}_c)) - T(\mathbf{x}))$ with $(T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x},\mathbf{p}_c))$. This entails a change of sign of $\delta\mathbf{p}$, so that, when updating the parameters estimate, the value of $\delta\mathbf{p}$ will be subtracted from the current parameters estimate rather than added. To compute $\delta\mathbf{p}$ efficiently, we assume that:

$$\left(\frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p}_c)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}} \Big|_{\mathbf{p}=\mathbf{p}_c} = \Gamma(\mathbf{x})\Sigma(\mathbf{p}_c) \tag{28}$$

where $\Gamma(\mathbf{x})$ is a $2 \times k$ matrix that is only function of the pixels coordinates on the template and $\Sigma(\mathbf{p}_c)$ is a $k \times n$ matrix depending on the current estimate of the parameters, for some integer $k > 0$. Then, matrix $H(\mathbf{p}_c)$ can be re-written as:

$$H(\mathbf{p}_c) = \Sigma(\mathbf{p}_c)^T H_* \Sigma(\mathbf{p}_c)\,, \tag{29}$$

where:

$$H_* = \sum_{\mathbf{x}} (\nabla T(\mathbf{x})\Gamma(\mathbf{x}))^T (\nabla T(\mathbf{x})\Gamma(\mathbf{x}))\,. \tag{30}$$

For simplicity's sake, we assume here that $k = n$ and that $\Sigma(\mathbf{p}_c)$ is invertible (see [5] for the general case $k \neq n$); then, the inverse of $H(\mathbf{p}_c)$ becomes:

$$H^{-1}(\mathbf{p}_c) = \Sigma(\mathbf{p}_c)^{-1} H_*^{-1} \Sigma(\mathbf{p}_c)^{-T}\,; \tag{31}$$

and the expression in Equation (27) reduces to:

$$\delta\mathbf{p} = \Sigma(\mathbf{p}_c)^{-1} H_*^{-1} \sum_{\mathbf{x}} (\nabla T(\mathbf{x})\Gamma(\mathbf{x}))^T (I(\mathbf{W}(\mathbf{x},\mathbf{p}_c) - T(\mathbf{x}))\,. \tag{32}$$

This formula yields an update of the parameters with a lower computational complexity than the forward algorithms, but under the very restrictive assumption that the composition of Equation (28) can be explicitely computed.

Parameters update: After computing $\delta\mathbf{p}$ with Equation (32), the current estimate of the warp is updated with the additive rule:

$$\mathbf{p}_{c+1} = \mathbf{p}_c - \delta\mathbf{p}, \tag{33}$$

where the minus is due to the change of sign introduced in Equation (27).

Assumptions on the set of warps: We assume that the warps $\mathbf{W} \in \mathcal{F}$ are differentiable; moreover, in order to find an expression for H_* that does not depend on $\mathbf{p}_c$, one has to explicitely find the decomposition of Equation (28). Moreover, in the general case, not only it is difficult to find an explicit decomposition, but it is not even obvious that it exists. This makes the IA algorithm usable with only a very limited set of warps in practical cases. In practice, authors of [5] show that IA algorithm can be employed with 2D translations, 2D affine warps and "a small number of esoteric non-linear warps" [2].

Computational complexity: The main steps of the IA algorithms are reported in Algorithm 4. The computational complexity of each step is reported as a function of the number of parameters n and the number of pixels of the template N, supposing for sake of simplicity that $k = n$.

Algorithm 4 Inverse Additional algorithm	
Pre-compute $(\nabla T(\mathbf{x})\Gamma(\mathbf{x}))$ for all the pixels of the template	$\mathcal{O}(nN)$
Pre-compute H_* using Equation (30)	$\mathcal{O}(nN)$
Pre-compute H_*^{-1}	$\mathcal{O}(n^3)$
while $\|\|\delta\mathbf{p}\|\| > \epsilon$ **do**	
Evaluate ${\Sigma\mathbf{p}_c}^{-1}$	$\mathcal{O}(n^2)$
Compute $\delta\mathbf{p}$ with Equation (27)	$\mathcal{O}(nN + n^2)$
Update the parameters using Equation (33)	$\mathcal{O}(n)$
end while	

Assuming that evaluating the matrix $\Sigma(\mathbf{p})^{-1}$ has a computational complexity of $\mathcal{O}(n^2)$, then the computational complexity of an iteration of the IA algorithm is:

$$\mathcal{O}(nN + n^2). \tag{34}$$

5 A second-order method: ESM

All the iterative methods described above are based on a first-order development of the terms appearing in the objective function. A priori, a second-order development should yield a more accurate approximation, but computing the second order derivatives is computationally too expensive for most applications. The Efficient Second-order Method (ESM) proposed in [3] allows to iteratively minimize a second-order approximation of the objective function using only first-order derivatives.

We start from same objective function as in the FC algorithm:

$$F_{ESM}(\delta\mathbf{p}) = F_{FC}(\delta\mathbf{p}) = \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c)) - T(\mathbf{x}))^2 \,;$$

Approximate solution: We perform the following *second-order* development around $\mathbf{p} = \mathbf{0}$:

$$\sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c)) - T(\mathbf{x}))^2 \approx \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) + \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)\delta\mathbf{p} + \frac{1}{2}\delta\mathbf{p}^T\mathbf{M}\delta\mathbf{p} - T(\mathbf{x}))^2 \,, \tag{35}$$

where:

$$\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c) = \frac{\partial}{\partial \mathbf{q}} \Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c)) \Big) \Big|_{\mathbf{q}=\mathbf{0}} \tag{36}$$

$$\mathbf{M} = \mathbf{M}(\mathbf{x}, \mathbf{p}_c) = \frac{\partial^2}{\partial \mathbf{q}^2} \Big(I(\mathbf{W}((\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c)) \Big) \Big|_{\mathbf{q}=\mathbf{0}} . \tag{37}$$

Now, let us assume that the following property holds for all the warps $\mathbf{W} \in \mathcal{F}$:

$$\begin{array}{c} \exists \epsilon > 0 \text{ such that } \forall \delta\mathbf{p} \in \mathbb{R}^n, ||\delta\mathbf{p}|| < \epsilon, \text{ then:} \\ \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}) = \mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}) \qquad \forall \mathbf{p} \in \mathbb{R}^n . \end{array} \tag{38}$$

Moreover, we assume here that $I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \approx T(\mathbf{x})$ and that $\delta\mathbf{p}$ is the (unknown) parameters increment such that $I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c)) = T(\mathbf{x})$.

Under these assumptions, the template jacobian $\frac{\partial}{\partial \mathbf{q}} \Big(T(\mathbf{W}(\mathbf{x}, \mathbf{q})) \Big) \Big|_{\mathbf{q}=\mathbf{0}} = \mathbf{J}_{IC}(\mathbf{x})$ can be approximated with a first-order development of the image jacobian $\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)$ around $\mathbf{p} = \mathbf{p}_c$, yielding:

$$\mathbf{J}_{IC}(\mathbf{x}) \approx \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c) + \mathbf{M}\delta\mathbf{p} . \tag{39}$$

In fact we have:

$$T(\mathbf{x}) \;=\; I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c)) \quad \Leftrightarrow$$

$$T(\mathbf{x}) \;=\; I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c + \delta\mathbf{p})) \quad \Leftrightarrow$$

$$T(\mathbf{W}(\mathbf{x}, \mathbf{0})) \;=\; I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{0}), \mathbf{p}_c + \delta\mathbf{p})) \quad \Leftrightarrow$$

$$\mathbf{J}_{IC}(\mathbf{x}) \;=\; \frac{\partial}{\partial \mathbf{q}} \Big(T(\mathbf{W}(\mathbf{x}, \mathbf{q})) \Big) \Big|_{\mathbf{q}=\mathbf{0}} = \frac{\partial}{\partial \mathbf{q}} \Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c + \delta\mathbf{p})) \Big) \Big|_{\mathbf{q}=\mathbf{0}} \quad \Leftrightarrow$$

$$\mathbf{J}_{IC}(\mathbf{x}) \;\approx\; \frac{\partial}{\partial \mathbf{q}} \Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c)) \Big) \Big|_{\mathbf{q}=\mathbf{0}} + \frac{\partial}{\partial \mathbf{p}} \frac{\partial}{\partial \mathbf{q}} \Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}) \Big) \Big|_{\mathbf{p}=\mathbf{p}_c,\ \mathbf{q}=\mathbf{0}} \delta\mathbf{p} \quad \Leftrightarrow$$

$$\mathbf{J}_{IC}(\mathbf{x}) \;\approx\; \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c) + \mathbf{M}\delta\mathbf{p},$$

From Equation (39) we compute the following approximation:

$$\mathbf{M}\delta\mathbf{p} \approx \mathbf{J}_{IC}(\mathbf{x}) - \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c); \tag{40}$$

by employing it into the second-order development of the objective function of Equation (35), we obtain a second-order approximation of the objective function, which can be computed using only first-order derivatives of the warp:

$$\sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c)) - T(\mathbf{x}))^2 \approx \sum_{\mathbf{x}} (I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) + \mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c)\delta\mathbf{p} - T(\mathbf{x}))^2 , \tag{41}$$

where:

$$\mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c) = \frac{\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c) + \mathbf{J}_{IC}(\mathbf{x})}{2} = \left(\frac{\nabla I_{\mathbf{p}_c}(\mathbf{x}) + \nabla T(\mathbf{x})}{2} \right) \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} \Big|_{\mathbf{p}=\mathbf{0}} . \quad (42)$$

The minimum of the quadratic form of Equation (41) is reached for:

$$\delta \mathbf{p} = H^{-1}(\mathbf{p}_c) \sum_{\mathbf{x}} \mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c)(I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) - T(\mathbf{x})) ; \quad (43)$$

where $H(\mathbf{p}_c) = \sum_{\mathbf{x}} \mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c)^T \mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c)$.

Parameters update: After computing a $\delta \mathbf{p}$ minimizing an approximation of $F_{ESM}(\delta \mathbf{p})$ based on a *second order development*, we can update the warp as in the FC algorithm, using Equation (15):

$$\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1}) = \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta \mathbf{p}), \mathbf{p}_c) . \quad (44)$$

Assumptions on the set of warps: The same assumptions as the FC algorithm should hold, more in particular that the warps $\mathbf{W} \in \mathcal{F}$ are differentiable, that the identity is an admissible warp and that $\mathcal{F}$ is closed with respect to the composition.

The assumption that the current estimate of the parameters is approximately exact and that $I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \approx T(\mathbf{x})$ guarantees that the finite order development of Equation (39) is valid, and that the parameters increment is the same as the one in Equation (35).

Moreover, in order to write the ESM parameters update equations, the cumbersome assumption (38) has to hold. Notice that this assumption does not only depend on the family of warps, but also on the parametrization chosen. If assumption (38) does not hold but $\mathcal{F}$ is a group of differentiable warps, the formula provided by ESM is still correct up to the first-order, as shown in Appendix C.

Computational complexity: The main steps of the ESM and their computational complexities are resumed in Algorithm 5.

Algorithm 5 Efficient Second-order Method

Pre-compute $\frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} \Big\|_{\mathbf{p}=\mathbf{0}}$, for all $\mathbf{x} \in \mathcal{D}$	$\mathcal{O}(nN)$
Pre-compute ∇T	$\mathcal{O}(N)$
while $\|\|\delta \mathbf{p}\|\| > \epsilon$ **do**	
Compute $\mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c)$ for all $\mathbf{x} \in \mathcal{D}$ with Equation (42)	$\mathcal{O}(nN)$
Compute $H(\mathbf{p}_c) = \sum_{\mathbf{x}} \mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c)^T \mathbf{J}_{ESM}(\mathbf{x}, \mathbf{p}_c)$	$\mathcal{O}(n^2 N)$
Compute $H(\mathbf{p}_c)^{-1}$	$\mathcal{O}(n^3)$
Compute $\delta \mathbf{p}$ with Equation (43)	$\mathcal{O}(nN + n^2)$
Update the parameters using Equation (44)	$\mathcal{O}(N)$
end while	

So, the computational complexity of one iteration of the ESM is the same as the forward algorithms:

$$\mathcal{O}(n^2 N + n^3) , \quad (45)$$

while, if the family of warps respects the assumptions specified above, ESM provides a more precise parameters update, thus converging in less iterations.

6 Choice of the appropriate algorithm. Additive vs Compositional approach

The algorithms described above mainly differ for their computational cost, for the assumptions made on the set of warps and for the accuracy of the approximation of the objective function. While in general cases each algorithm computes a different parameters update, it can be shown that the 4 first-order algorithms are equivalent [2], in the following sense:

> at a given iteration c, the 4 first-order algorithms provide the same updated warp $\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1})$, up to a first-order development in the second argument of the warp.

In Appendix A we show the equivalence of FA, FC and IC (we don't treat the case of IA since it's limited interest for practical applications, the interested reader may refer to [2]). We observe that ESM is equivalent to the other algorithms in the sense specified above, since it provides the same update as FC up to the first order.

The choice of the correct algorithm for a practical problem depends, among others, on the following factors:

- assumptions on the set of warps $\mathcal{F}$;
- comparison of the computational complexity for one iteration;
- comparison of the computational complexity of the updated parameters estimate.

If Assumption (38) holds, then ESM should converge in less iterations at a computational cost comparable with that of the first-order forward algorithms. Unfortunately, this seldom happens, with the important exception of the family of homographies (with an opportune parametrization).

As for the first-order methods, since usually FC does not improve much the computational efficiency of FA, the choice is made between FA anc IC algorithms. The latter has a better computational complexity, but it requires to compute the inverse warp and to find an explicit update of the parameters starting from the updated warp, so the effective computational cost should be compared in practical cases.

Another important difference is that, while computing the matrix $\mathbf{J}$, FA uses the gradient of the image, while IC uses the gradient of the template. If for some reason, one between I and T is much more affected by noise than the other, one should choose the algorithm that allows for the less noisy computation of $\mathbf{J}$.

Finally, we observe that, despite the fact that the additive and compositional algorithms yield to equivalent results, they reflect two different ways of interpreting the problem of image alignment, as depicted in Figure 5. In the first one, the "additive" point of view, a pixel $\mathbf{x}$ in the system of reference of the template is progressively warped on the image for finding a pixel with the same intensity. In the "compositional" point of view, at each iteration we compare 2 images $T(\mathbf{x})$ and $I_{\mathbf{p}}(\mathbf{x})$ in the same reference system, and iteratively look for an infinitesimal warp $\mathbf{W}(\mathbf{x}, \delta\mathbf{p})$ such that either $I_{\mathbf{p}}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}))$ is closer to $T(\mathbf{x})$ (in FC), or $T(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}))$ is closer to $I_{\mathbf{p}}(\mathbf{x})$ (in IC).

This can be done by comparing the formulas of the compositional algorithms, FC and IC. The first term of the jacobian matrix of the FC algorithm reported in Equation (13) corresponds to the gradient of the warped image $I_{\mathbf{p}_c}$ defined in Equation (3)

$$\nabla I(\mathbf{W}(\mathbf{x}, \mathbf{p}_c)) \frac{\partial \mathbf{W}(\mathbf{y}, \mathbf{p}_c)}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{x}} = \nabla I_{\mathbf{p}_c}(\mathbf{x}). \tag{46}$$

So, $\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)$ can alternatively be computed as:

$$\mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c) = \nabla I_{\mathbf{p}_c}(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{0}}. \tag{47}$$

This expression is very close to that of $\mathbf{J}_{IC}(\mathbf{x})$ reported in Equation (19):

$$\mathbf{J}_{IC}(\mathbf{x}) = \nabla T(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{0}}. \tag{48}$$

Moreover, comparing the formulas for the computation of $\delta\mathbf{p}$ for FC (Equation (14)):

$$\delta\mathbf{p} = H(\mathbf{p}_c)^{-1} \sum_{\mathbf{x}} \mathbf{J}_{FC}(\mathbf{x}, \mathbf{p}_c)^T (T(\mathbf{x}) - I_{\mathbf{p}_c}(\mathbf{x})) , \tag{49}$$

and for IC (Equation (18)):

$$\delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \mathbf{J}_{IC}(\mathbf{x})^T (I_{\mathbf{p}_c}(\mathbf{x}) - T(\mathbf{x})) , \tag{50}$$

it is possible to see that at a given iteration, the computation of $\delta\mathbf{p}$ is performed exactly in the same way in the 2 compositional algorithms, except that the roles of $T(\mathbf{x})$ and $I_{\mathbf{p}_c}(\mathbf{x})$ are switched. This does not mean that the computed values of $\delta\mathbf{p}$ are the same for the 2 algorithms, they only provide equivalent warps, as shown in Appendix A.

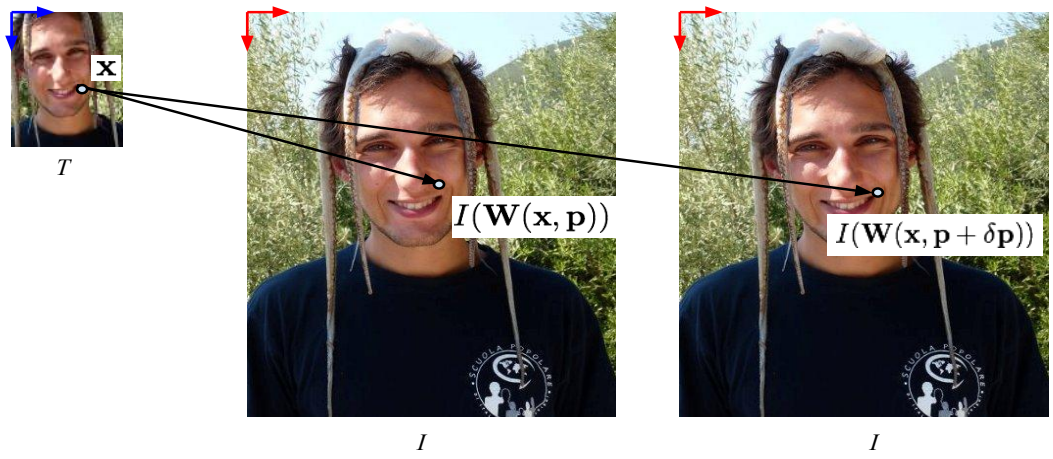
7 Some examples of warps

In order to illustrate the differences among the alignment methods introduced above, in this section we show 2 practical examples of warps. The first is a rigid, 2D warp, while the second is a highly non-linear warp involving a 3D perspective transformation. Other examples (affine warps, homographies) are reported in [2].

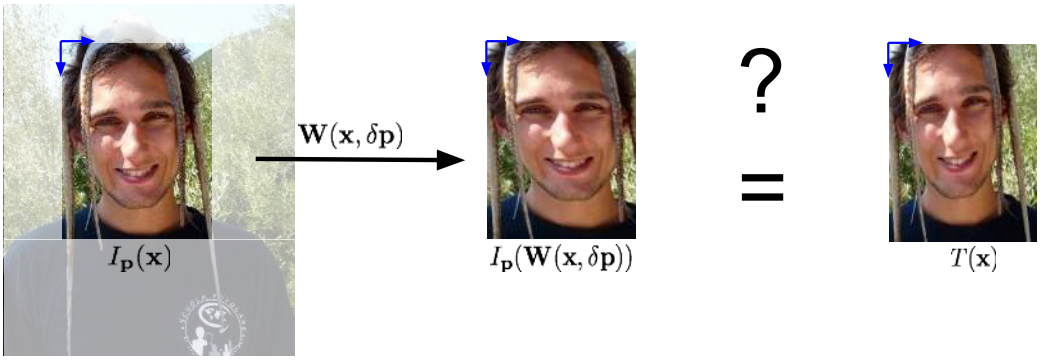
7.1 Rigid 2D warp

Let $\mathcal{F}$ be the family of rigid transforms of the plane parametrized by an array of 3 parameters $\mathbf{p} \in \mathbb{R}^3$. At iteration c, given the current parameters $\mathbf{p}_c = (\theta, t_1, t_2)^T$, the warp for a pixel $\mathbf{x}$ is computed as:

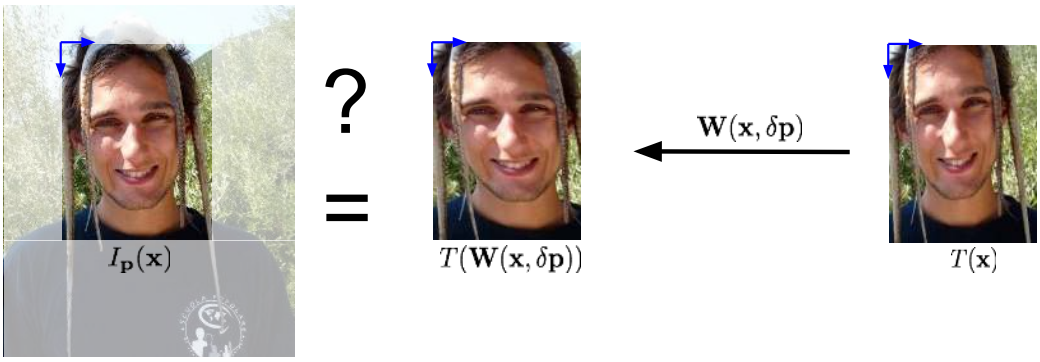
$$\mathbf{W}(\mathbf{x}, \mathbf{p}_c) = \mathbf{R}_{\mathbf{p}_c}\mathbf{x} + \mathbf{t}_{\mathbf{p}_c} = \begin{bmatrix} \cos(\theta) & -sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} . \tag{51}$$



(a)



(b)



(c)

Figure 5: Schematic representation of dense image alignment algorithms. (a): FA algorithm. (b): FC algorithm (c) IC algorithm.

the family of warps is closed with respect to the composition and to the inversion. The warps are differentiable with respect to all the arguments; the derivative with respect to the parameters is given by:

$$\frac{\partial}{\partial \mathbf{p}}\mathbf{W}(\mathbf{x},\mathbf{p}) = \begin{bmatrix} -\sin(\theta)u + \cos(\theta)v & 1 & 0 \\ -\cos(\theta)u - \sin(\theta)v & 0 & 1 \end{bmatrix} , \tag{52}$$

while $\frac{\partial}{\partial \mathbf{x}}\mathbf{W}(\mathbf{x},\mathbf{p}) = \mathbf{R}_{\mathbf{p}}$. Finally, we note that $\mathbf{W}(\mathbf{x},\mathbf{0}) = \mathbf{x}$.

Forward Additive algorithm: The assumptions for FA algorithm hold. Once the parameters increment $\delta\mathbf{p} = (\delta\theta, \delta t_1, \delta t_2)$ has been computed, the updated warp is given by:

$$\mathbf{W}(\mathbf{x},\mathbf{p}_{c+1}) = \begin{bmatrix} \cos(\theta+\delta\theta) & -\sin(\theta+\delta\theta) \\ \sin(\theta+\delta\theta) & \cos(\theta+\delta\theta) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} t_1+\delta t_1 \\ t_2+\delta t_2 \end{bmatrix} , \tag{53}$$

Forward Compositional algorithm: Assumptions of FC algorithm are satisfied. Once the parameters increment $\delta\mathbf{p} = (\delta\theta, \delta t_1, \delta t_2)$ has been computed, the updated warp is given by:

$$\mathbf{W}(\mathbf{x},\mathbf{p}_{c+1}) = \mathbf{R}_{\mathbf{p}_c}(\mathbf{R}_{\delta\mathbf{p}}\mathbf{x} + \mathbf{t}_{\delta\mathbf{p}}) + \mathbf{t}_{\mathbf{p}_c} ; \tag{54}$$

After computing the updated rotation matrix $\mathbf{R}_{\mathbf{p}_{c+1}} = \mathbf{R}_{\mathbf{p}_c}\mathbf{R}_{\delta\mathbf{p}}$ and and the updated translation vector $\mathbf{t}_{\mathbf{p}_{c+1}} = \mathbf{R}_{\mathbf{p}_c}\mathbf{t}_{\delta\mathbf{p}} + \mathbf{t}_{\mathbf{p}_c}$, we can explicitly estimate the updated parameters as, for example:

$$\mathbf{p}_{c+1} = \begin{bmatrix} \cos^{-1}(R^{11}_{\mathbf{p}_{c+1}}) \\ \mathbf{t}_{\mathbf{p}_{c+1}} \end{bmatrix} , \tag{55}$$

Inverse Compositional algorithm: the inverse of a rigid warp is a rigid warp; the inverse warp is given by:

$$\mathbf{W}^{-1}(\mathbf{x},\mathbf{p}) = \mathbf{R}^T_{\mathbf{p}}(\mathbf{x} - \mathbf{t}_{\mathbf{p}}) . \tag{56}$$

At iteration c, once the parameters increment $\delta\mathbf{p} = (\delta\theta, \delta t_1, \delta t_2)$ has been computed, the updated warp is given by:

$$\mathbf{W}(\mathbf{x},\mathbf{p}_{c+1}) = \mathbf{R}_{\mathbf{p}_c}(\mathbf{R}^T_{\delta\mathbf{p}}(\mathbf{x} - \mathbf{t}_{\delta\mathbf{p}})) + \mathbf{t}_{\mathbf{p}_c} , \tag{57}$$

and we can explicitly update the parameters in an analogous way as the FC algorithm.

Inverse Additive algorithm: In order to apply Inverse Additive algorithm, we have to explicitly compute decomposition of Equation (28). Since:

$$\left(\frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p}_c)}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} = \begin{bmatrix} \cos(\theta) & sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} -\sin(\theta)u + \cos(\theta)v & 1 & 0 \\ -\cos(\theta)u - \sin(\theta)v & 0 & 1 \end{bmatrix} , \tag{58}$$

it is not possible to decompose the above Equation in the product of 2 matrices $\Gamma(\mathbf{x})\Sigma(\mathbf{p})$, and thus the Inverse Additive algorithm can not be employed, even with this very simple warp and this parametrization.

Efficient Second-order method: As can be observed comparing the warp updated with the additional rule (Equation (53)) and that updated with the compositional rule (Equation (54)), assumption (38) does not hold in this case, so ESM does not minimize a second-order approximation of the objective function. Nonetheless, since the warps are differentiable and form a group, ESM will provide a first-order method (see Appendix C for more details).

7.2 A warp for 3D tracking

An interesting family of warps is the one shown in Figure 6, since it allows to estimate the 3D pose of an image in a general, non-planar scene by aligning it against a template with known pose. We suppose the mapping between a point $\mathbf{X} \in \mathbb{R}^3$ in the 3D world reference system and its representation $\mathbf{x} \in \mathbb{R}^2$ on a picture is given by the standard projective model: $\mathbf{x} = \mathcal{P}(\mathbf{X})$, where the projection is computed as:

$$\tilde{x} = (u, v, z)^T = \mathbf{K}(\mathbf{R}\mathbf{X} + \mathbf{T}) \,,$$
$$\mathbf{x} = (u/z, v/z)^T \,.$$

In this context, the template T is given by an image showing a 3D scene with a known pose $\mathbf{p}_T$. The parameters $\mathbf{p} \in \mathbb{R}^6$ of the warp shown in Figure 6 describe the pose of camera for another image I showing the same 3D scene (the actual pose of I is given by $\mathbf{p}_T + \mathbf{p}$, so that $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$).

We suppose all the internal parameters of both images are known. So, the pose of the image can be estimated applying the methods described above for aligning T and I through the warp of Figure 6. In order to compute the warp, the 3D structure of the scene (e.g. a 3D model) must be known.

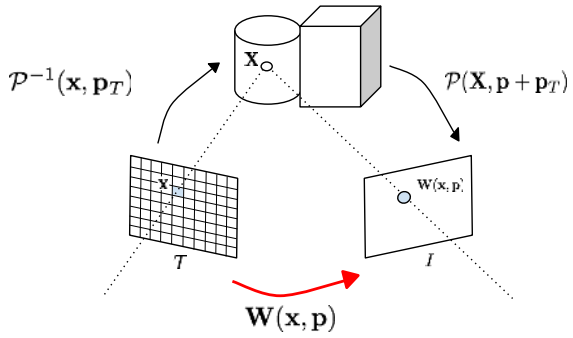


Figure 6: 3D warp between a template T with pose $\mathbf{p}_T$, and an image I, with pose $\mathbf{p}_T + \mathbf{p}$.

The inverse warp is depicted in Figure 7.

The warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ of Figure 6 is not continuous with respect to the first argument in regions close to occluding contours, while, for a given pixel $\mathbf{x}$, it is differentiable with respect to $\mathbf{p}$, since

$$\frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} = \frac{\partial \mathcal{P}(\mathbf{X}, \mathbf{p})}{\partial \mathbf{p}} \,. \tag{59}$$

The composition of warps is easily computable, but the warp is closed with respect to composition only for planar scenes, not in a general case. Moreover, $\mathbf{W}$ is only piece-wise

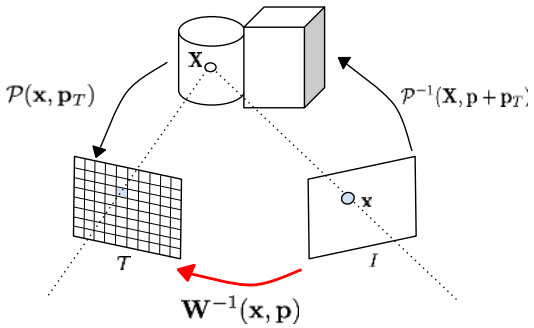


Figure 7: Inverse warp for 3D tracking.

differentiable with respect to the first argument, since its spatial derivative $\frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p})}{\partial \mathbf{x}}$ is well-defined only for pixels far from occluding contours and representing locally smooth surfaces of the scene.

Moreover, although the warp is mathematically well-defined for all the pixels of T, a pixel $\mathbf{x}$ on T and its corresponding pixel $\mathbf{W}(\mathbf{x}, \mathbf{p}_I)$ on I may not represent the same 3D point $\mathbf{X} = \mathcal{P}^{-1}(\mathbf{x}, \mathbf{p}_T)$ if $\mathbf{X}$ is not visible in both images; in these cases, the inverse warp does not exist, either.

It is possible to check what are the pixels for which $\mathbf{W}^{-1}$ is well defined, by checking that $\mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{p}_c), \mathbf{p}_c)^{-1} = \mathbf{x}$, but in practical cases this is un-necessary. If the the poses of T and I are not too far, the direct and inverse warp are well defined for most part of the pixels, and the influence of badly warped pixels is limited.

If the template and the image share the same internal calibration matrix then $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$. The generalization to the case of different internal calibration matrices is treated in Section 7.3.

For the compositional algorithms, at each iteration an explicit estimate of the updated parameters $\mathbf{p}_{c+1}$ is needed, that can not be analytically computed starting from the $\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1})$. In practical cases, it is possible to compute the updated warp $\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1})$ with Equation (15) or (20) for a subset of the pixels of the template; then, since $\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1}) = \mathcal{P}(\mathbf{X}, \mathbf{p}_T + \mathbf{p}_{c+1})$, an explicit estimate of $\mathbf{p}_{c+1}$ can be computed from the 3D-2D correspondences $\mathbf{X} \leftrightarrow \mathcal{P}(\mathbf{X}, \mathbf{p}_T + \mathbf{p}_{c+1})$ using, for instance, a PnP algorithm.

Notice that, when computing the updated warp for the IC algorithm with Equation (20), the following simplification holds:

$$\begin{aligned}
&\mathbf{W}(\mathbf{W}^{-1}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c) \\
=&\mathcal{P}(\mathcal{P}^{-1}(\mathcal{P}(\mathcal{P}^{-1}(\mathbf{x}, \mathbf{p}_T + \delta\mathbf{p}), \mathbf{p}_T), \mathbf{p}_T), \mathbf{p}_T + \mathbf{p}_c) \\
=&\mathcal{P}(\mathcal{P}^{-1}(\mathbf{x}, \mathbf{p}_T + \delta\mathbf{p}), \mathbf{p}_T + \mathbf{p}_c)\,.
\end{aligned}$$

The IA algorithm is not employable with $\mathbf{W}$, since no decomposition of the form of Equation (28) is easilly computable.

As for application of ESM algorithm, assumption of Equation (38) does not hold, we introduce an error in the second order terms, and ESM just provides a first-order method

(see Appendix C for more details). Nonetheless, we experimentally observed that applying ESM to 3D tracking often entails a benefice, since, putting an average of the gradients of the template and the image in the jacobian matrix can reduce the influence of the noise in the images.

7.3 3D tracking with different internal matrices

A fundamental hypothesis for employing the FC, the IC and the ESM algorithms is that $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$. That is true for the warp of Figure 6, only if the image and the template have the same internal calibration matrix. If T and I have different internal calibration matrices (say, respectively, $\mathbf{K}_T$ and $\mathbf{K}_{IM}$), one can pre-warp the image and use $\tilde{I}(\mathbf{x}) = I(\mathbf{K}_T \mathbf{K}_{IM}^{-1} \mathbf{x})$. [1] Alternatively, as shown in Figure 8, it is possible to split the global warp in 2 parts, the warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ estimated through image alignment, that employs exclusively the template internal calibration matrix $\mathbf{K}_T$ (in the red box in Figure 8), and an additional transformation $\widehat{\mathbf{W}(\mathbf{x}, \mathbf{p})} = \mathbf{K}_{IM} \mathbf{K}_T^{-1} \mathbf{W}(\mathbf{x}, \mathbf{p})$ for reading intensity values on the image. From the implementation point of view, this means there is no need of pre-warping the image, but only:

1. employ $\mathbf{K}_T$ to compute the warp derivatives and the jacobian matrices of the 3D warp;
2. use $I(\widehat{\mathbf{W}(\mathbf{x}, \mathbf{p})})$ instead of $I(\mathbf{W}(\mathbf{x}, \mathbf{p}))$ for retrieving the luminous intensity values of the image pixels. Notice that

$$I(\mathbf{K}_{IM} \mathbf{K}_T^{-1} \mathcal{P}_{\mathbf{K}_T}(\mathbf{X}, \mathbf{p}_T + \mathbf{p})) = I(\mathcal{P}_{\mathbf{K}_{IM}}(\mathbf{X}, \mathbf{p}_T + \mathbf{p})). \tag{60}$$

8 Appendix A : Equivalence of first-order methods

The equivalence of the 4 first-order methods described above (FA, FC, IC and IA) has been shown in [2], in the sense that, at a given iteration, all the algorithms provide the same update for the warp, up to a first order development in $\delta\mathbf{p}$.

We show here only the equivalence of FA and FC algorithms, and that of FC and IC algorithms, therefore showing that the 3 algorithms are all equivalent. Even if the same result holds for IA algorithm, we don't report demonstration here since IA is of little interest for most part of applications (the interested reader may refer to [2] for further details).

8.1 Equivalence of FA and FC algorithms

We can approximate the updated warp sought at each iteration of the FA algorithm with the following first-order development:

$$\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1}) = \mathbf{W}(\mathbf{x}, \mathbf{p}_c + \delta\mathbf{p}) \approx \mathbf{W}(\mathbf{x}, \mathbf{p}_c) + \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} \delta\mathbf{p}. \tag{61}$$

[1] For sake of simplicity, in this section we employ a slight abuse of notation employing 2D quantities such as $\mathbf{x}$ and $\mathbf{W}(\mathbf{x}, \mathbf{p})$ and the corresponding quantities in homogeneous coordinates, respectively $\tilde{\mathbf{x}} = [\mathbf{x}\ 1]^T$, $\tilde{\mathbf{W}}(\mathbf{x}, \mathbf{p}) = [\mathbf{W}(\mathbf{x}, \mathbf{p})\ 1]^T$, interchangeably, since this does not lead to confusion.

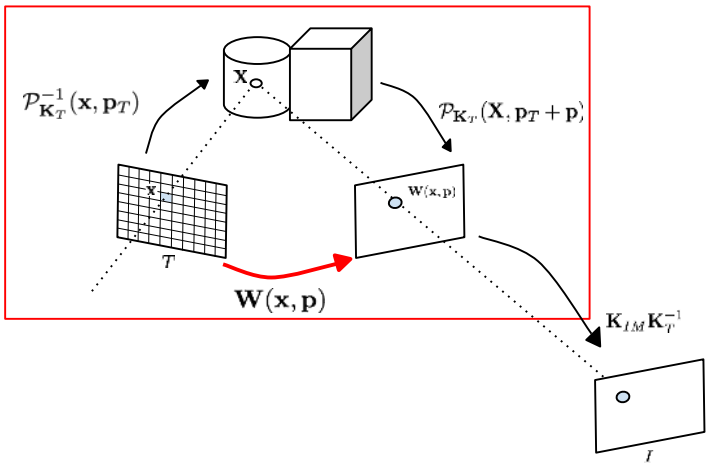


Figure 8: Warp between a template T with internal calibration matrix $\mathbf{K}_T$ and pose $\mathbf{p}_T$, and an image I, with pose $\mathbf{p}_T + \mathbf{p}$ and internal calibration matrix $\mathbf{K}_{IM}$. Dependence of the projective transforms on the internal calibration matrix has been highlighted: $\mathcal{P}_{\mathbf{K}}(\mathbf{X}, \mathbf{p}) = \mathbf{K}(\mathbf{R}(\mathbf{p})\mathbf{X} + \mathbf{t}(\mathbf{p}))$

As for the FC algorithm, an analogous approximation gives:

$$\mathbf{W}(\mathbf{x}, \mathbf{p}_{c+1}) = \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}_c) \approx \mathbf{W}(\mathbf{x}, \mathbf{p}_c) + \frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c)}{\partial \mathbf{q}}\Big|_{\mathbf{q}=\mathbf{0}} \delta\mathbf{p}. \tag{62}$$

So, the values of $\delta\mathbf{p}$ minimizing $F_{FA}(\delta\mathbf{p})$ and $F_{FC}(\delta\mathbf{p})$ are the same (up to the first order) if $\frac{\partial \mathbf{W}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c}$ and $\frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x},\mathbf{q}),\mathbf{p}_c)}{\partial \mathbf{q}}\Big|_{\mathbf{q}=\mathbf{0}}$ share the same linear space, that is, if there is an invertible matrix $A \in \mathbb{R}^{2\times 2}$ such that:

$$\frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} = A \, \frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c)}{\partial \mathbf{q}}\Big|_{\mathbf{q}=\mathbf{0}}. \tag{63}$$

Actually, such matrix exists if the warps are differentiable and closed with respect to inversion and composition (see Appendix B for a proof).

Therefore, since the optimal update is sought in the same linear space, the updates of the warps computed by FA and FC at a given iteration are the same up to a first-order development in $\delta\mathbf{p}$.

8.2 Equivalence of FC and IC algorithms

We start by interpreting the sum Equation (11) as an integral over the domain of the template $\mathcal{D}$:

$$F_{FC}(\delta\mathbf{p}) = \int_{\mathcal{D}} \Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p})) - T(\mathbf{x})\Big)^2 d\mathbf{x}. \tag{64}$$

The change of variables $\mathbf{y} = \mathbf{W}(\mathbf{x}, \delta\mathbf{p})$ gives:

$$F_{FC}(\delta\mathbf{p}) = \int_{\mathbf{W}(\mathcal{D},\delta\mathbf{p})} \Big(I(\mathbf{W}(\mathbf{y}, \mathbf{p})) - T(\mathbf{W}^{-1}(\mathbf{y}, \delta\mathbf{p}))\Big)^2 \Big|\frac{\partial \mathbf{W}^{-1}(\mathbf{y}, \delta\mathbf{p})}{\partial \mathbf{y}}\Big| d\mathbf{y}. \tag{65}$$

We observe that $\mathbf{W}(\mathcal{D}, \delta\mathbf{p}) \approx \mathcal{D}$ up to a zero-th order, and that

$$\Big|\frac{\partial \mathbf{W}^{-1}(\mathbf{y}, \delta\mathbf{p})}{\partial \mathbf{y}}\Big| = 1 + \mathcal{O}(\delta\mathbf{p}), \tag{66}$$

since $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$. Making the assumption that $(I(\mathbf{W}(\mathbf{y}, \mathbf{p})) - T(\mathbf{W}^{-1}(\mathbf{y}, \delta\mathbf{p})))$ (or, equivalently, $(I(\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p})) - T(\mathbf{x}))$) is $\mathcal{O}(\delta\mathbf{p})$, we can approximate Equation (65) up to the first order ignoring the higher order terms in $\delta\mathbf{p}$:

$$F_{FC}(\delta\mathbf{p}) \approx \int_{\mathcal{D}} \Big(I(\mathbf{W}(\mathbf{y}, \mathbf{p})) - T(\mathbf{W}^{-1}(\mathbf{y}, \delta\mathbf{p}))\Big)^2 d\mathbf{y}; \tag{67}$$

this expression is formally identical to $F_{IC}(\delta\mathbf{p})$ of Equation (17) (interpreting the sum as an integral over the domain of the template), except for the inverse warp in the template $T(\mathbf{W}^{-1}(\mathbf{y}, \delta\mathbf{p}))$. Since in the IC update rule of Equation (20) the warp of the template is inverted before composing it with the current warp, we conclude that the updated warps computed by FC and IC are equivalent up to a first order development in $\delta\mathbf{p}$.

9 Appendix B: A theorem about groups of differentiable warps

We prove here the following theorem, needed for demonstrating the equivalence of FA and FC algorithms in Appendix A:

Theorem 1. *Let $\mathcal{F}$ be a group of differentiable warps $\mathbf{W} : \mathbb{R}^2 \times \mathbb{R}^n \to \mathbb{R}^2$. Then, for any pixel $\mathbf{x} \in \mathbb{R}^2$, an invertible matrix $A \in \mathbb{R}^{2\times 2}$ exists (eventually depending on $\mathbf{x}$), such that:*

$$\frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} = A(\mathbf{x}) \; \frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c)}{\partial \mathbf{q}}\Big|_{\mathbf{q}=\mathbf{0}}. \tag{68}$$

In order to prove Theorem 1, we make use of the following:

Theorem 2. *Let $\mathbf{x} \in \mathbb{R}^2$ some fixed pixel, and $\mathcal{F}$ be a group of differentiable warps $\mathbf{W} : \mathbb{R}^2 \times \mathbb{R}^n \to \mathbb{R}^2$. Then a function $\phi : \mathbb{R}^n \to \mathbb{R}^n : \quad \delta\mathbf{p} \mapsto \delta\mathbf{p}' = \phi(\delta\mathbf{p})$ can be defined in some open ball aroud the origin $\mathcal{B}_\delta(\mathbf{0})$, such that:*

- $\phi(\mathbf{0}) = \mathbf{0}$;
- *ϕ is differentiable and invertible in $\mathcal{B}_\delta(\mathbf{0})$;*
- $\mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}) = \mathbf{W}(\mathbf{W}(\mathbf{x}, \phi(\delta\mathbf{p})), \mathbf{p}) \qquad \forall \mathbf{p} \in \mathbb{R}^n$;

Proof. First, we observe that there is a $\epsilon > 0$ such that, for all $\delta\mathbf{p} \in \mathbb{R}^n$, $\delta\mathbf{p} \in \mathcal{B}_\epsilon(\mathbf{0})$, there is $\delta\mathbf{p}' \in \mathbb{R}^n$ such that:

$$\mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}) = \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}'), \mathbf{p}) \qquad \forall \mathbf{p} \in \mathbb{R}^n; \tag{69}$$

Since $\mathcal{F}$ is closed under inversion and composition, $\mathbf{W}^{-1}(\mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}), \mathbf{p}) \in \mathcal{F}$, so there must be some $\delta\mathbf{p}'$ so that $\mathbf{W}(\mathbf{x}, \delta\mathbf{p}') := \mathbf{W}^{-1}(\mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}), \mathbf{p})$.

Then, we observe that,under the same assumptions, the inverse statement is also true, that is, there is a $\tilde{\epsilon} > 0$ such that, for all $\delta\mathbf{p} \in \mathbb{R}^n$, $||\delta\mathbf{p}|| < \tilde{\epsilon}$, there is $\delta\mathbf{p}' \in \mathbb{R}^n$ such that:

$$\mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}) = \mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}') \quad \forall \mathbf{p} \in \mathbb{R}^n \,. \tag{70}$$

Applying the Generalized Implicit Function Theorem ([4]) to the continuously differentiable function $F(\delta\mathbf{p}, \delta\mathbf{p}') = \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}) - \mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}')$, we deduce that a function $\phi : \mathbb{R}^n \mapsto \mathbb{R}^n : \quad \delta\mathbf{p} \mapsto \phi(\delta\mathbf{p}) = \delta\mathbf{p}'$ exists, which is differentiable and invertible in some open ball around $\mathbf{0}$ and such that $\phi(\mathbf{0}) = \mathbf{0}$. □

Now, we notice that, given a $\mathbf{p}, \mathbf{q}, \mathbf{r} \in \mathbb{R}^n$, we have :

$$\frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p} + \mathbf{q})}{\partial \mathbf{p}} = \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p} + \mathbf{q})}{\partial \mathbf{q}} \,; \tag{71}$$

and:

$$\frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{r} + \mathbf{q}), \mathbf{p})}{\partial \mathbf{r}} = \frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{r} + \mathbf{q}), \mathbf{p})}{\partial \mathbf{q}} \,. \tag{72}$$

Now, let $\delta\mathbf{p} \in \mathbb{R}^n$ small enough, so that we can define a function $\phi(\delta\mathbf{p})$ as in Theorem 2. We have:

$$\frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} = \frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p}_c + \delta\mathbf{p})}{\partial \delta\mathbf{p}} = \tag{73}$$

$$\frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}'), \mathbf{p}_c)}{\partial \delta\mathbf{p}'} \frac{\partial \delta\mathbf{p}'}{\partial \delta\mathbf{p}} \approx \nabla\phi(\delta\mathbf{p}) \frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q} + \delta\mathbf{p}'), \mathbf{p}_c)}{\partial \mathbf{q}}\Big|_{\mathbf{q}=\mathbf{0}}. \tag{74}$$

Finally, the statement of Theorem 1 is obtained by evaluating the above expression for $\delta\mathbf{p} = \mathbf{0}$. So, the invertible matrix A of Equation (68) is given by $\nabla\phi(\mathbf{0})$: this shows that $\frac{\partial \mathbf{W}(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c}$ and $\frac{\partial \mathbf{W}(\mathbf{W}(\mathbf{x}, \mathbf{q}), \mathbf{p}_c)}{\partial \mathbf{q}}\Big|_{\mathbf{q}=\mathbf{0}}$ share the same linear space, the tangent space of the manifold $\mathbf{W}(\mathbf{x}, \mathbf{p}_c)$.

10 Appendix C: relaxing hypothesis of ESM

In order to apply ESM to a family of warps $\mathcal{F}$, the cumbersome hypothesis of Equation (38) must hold, seriously limiting the practical applications of ESM. In this section we show that, if this hypothesis does not hold, but $\mathcal{F}$ is a group of differentiable warps, then ESM looses the second-order accuracy and becomes a first-order method as FC.

As shown in Section 5, ESM is built computing a second-order development of the objective function (Equation (35)) and replacing the second-order term of this expression with an approximation based on the first-order development of the image jacobian $\mathbf{J}_{FC}$ of Equation (39). If assumption (38) does not hold, the development of Equation (39) is no longer valid; however, it is possible to employ Theorem 2 for a quantitative estimate of the error done in ESM, showing that ESM objective function approximation is still accurate up to the first order.

Let $\mathcal{F}$ a group of differentiable warps and $\delta\mathbf{p} \in \mathbb{R}^n$ such that $||\delta\mathbf{p}||$ is small enough. Then, thanks to Theorem 2, there's $\delta\mathbf{p}' = \phi^{-1}(\delta\mathbf{p})$ such that:

$$\mathbf{W}(\mathbf{x}, \mathbf{p} + \delta\mathbf{p}') = \mathbf{W}(\mathbf{W}(\mathbf{x}, \delta\mathbf{p}), \mathbf{p}) \qquad \forall \mathbf{p} \in \mathbb{R}^n; \tag{75}$$

So, the development of Equation (39) becomes:

$$T(\mathbf{x}) \;=\; I(\mathbf{W}(\mathbf{W}(\mathbf{x},\delta\mathbf{p}),\mathbf{p}_c)) \quad \Leftrightarrow$$

$$T(\mathbf{x}) \;=\; I(\mathbf{W}(\mathbf{x},\mathbf{p}_c+\delta\mathbf{p}')) \quad \Leftrightarrow$$

$$T(\mathbf{W}(\mathbf{x},\mathbf{0})) \;=\; I(\mathbf{W}(\mathbf{W}(\mathbf{x},\mathbf{0}),\mathbf{p}_c+\delta\mathbf{p}')) \quad \Leftrightarrow$$

$$\mathbf{J}_{IC}(\mathbf{x}) \;=\; \frac{\partial}{\partial\mathbf{q}}\Big(T(\mathbf{W}(\mathbf{x},\mathbf{q}))\Big)\Big|_{\mathbf{q}=\mathbf{0}} = \frac{\partial}{\partial\mathbf{q}}\Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x},\mathbf{q}),\mathbf{p}_c+\delta\mathbf{p}'))\Big)\Big|_{\mathbf{q}=\mathbf{0}} \quad \Leftrightarrow$$

$$\mathbf{J}_{IC}(\mathbf{x}) \;\approx\; \frac{\partial}{\partial\mathbf{q}}\Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x},\mathbf{q}),\mathbf{p}_c))\Big)\Big|_{\mathbf{q}=\mathbf{0}} + \frac{\partial}{\partial\mathbf{p}}\frac{\partial}{\partial\mathbf{q}}\Big(I(\mathbf{W}(\mathbf{W}(\mathbf{x},\mathbf{q}),\mathbf{p}))\Big)\Big|_{\mathbf{p}=\mathbf{p}_c,\ \mathbf{q}=\mathbf{0}}\delta\mathbf{p}' \quad \Leftrightarrow$$

$$\mathbf{J}_{IC}(\mathbf{x}) \;\approx\; \mathbf{J}_{FC}(\mathbf{x},\mathbf{p}_c) + \mathbf{M}\delta\mathbf{p}' \Leftrightarrow$$

$$\mathbf{J}_{IC}(\mathbf{x}) \;\approx\; \mathbf{J}_{FC}(\mathbf{x},\mathbf{p}_c) + \mathbf{M}\nabla\phi^{-1}(\mathbf{0})\delta\mathbf{p}.$$

That is, using the approximation $\mathbf{M}\delta\mathbf{p} \approx \mathbf{J}_{IC}(\mathbf{x}) - \mathbf{J}_{FC}(\mathbf{x},\mathbf{p}_c)$ in Equation (35), we introduce an error:

$$\mathbf{e} = (\nabla\phi^{-1}(\mathbf{0}) - \mathbb{I}))\delta\mathbf{p} \tag{76}$$

in the second-order term, so that the approximation of the objective function is correct only up to the first order. If Assumption (38) hold, then ϕ is the identity function and the error is null.

Appendix D: Comparative Tables

Table 1 resumes all the algorithms described in this survey, along with their computational complexity, the order of approximation of the objective function and the hypothesis made on the family of parametrized warps.

Table 2 shows the formulas employed by each algorithm. All the algorithms seek for a parameters increment $\delta\mathbf{p}$, approximately minimizing an objective function $F(\delta\mathbf{p})$. The parameters increment $\delta\mathbf{p}$ is computed by all the methods as:

$$\delta\mathbf{p} = \alpha\ H^{-1}\sum_{\mathbf{x}}\mathbf{J}(\mathbf{x})^T\ \left(T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x},\mathbf{p}_c))\right), \tag{77}$$

where $H = \sum_{\mathbf{x}}(\mathbf{J}(\mathbf{x})^T\ \mathbf{J}(\mathbf{x}))$, and

$$\alpha = \begin{cases} 1 & \text{for forward algorithms (FA, FC, ESM)} \\ -1 & \text{for inverse algorithms (IA, IC)}. \end{cases}$$

Algo	Order of approx.	Computational Complexity	Assumptions on $\mathcal{F}$
FA	I	$\mathcal{O}(n^2N + n^3)$	$\mathbf{W}(\mathbf{x}, \mathbf{p})$ differentiable wrt $\mathbf{p}$
FC	I	$\mathcal{O}(n^2N + n^3)$	$\mathbf{W}(\mathbf{x}, \mathbf{p})$ differentiable. $\mathcal{F}$ is a semi-group
IC	I	$\mathcal{O}(n^2N + n^2)$	$\mathbf{W}(\mathbf{x}, \mathbf{p})$ differentiable. $\mathcal{F}$ is a group
IA	I	$\mathcal{O}(n^2N + n^2)$	$\mathbf{W}(\mathbf{x}, \mathbf{p})$ differentiable. Decomposition of Eq. (28)
ESM	II	$\mathcal{O}(n^2N + n^3)$	$\mathbf{W}(\mathbf{x}, \mathbf{p})$ differentiable. $\mathcal{F}$ is a semi-group. Hyp. of Eq. (38)

Table 1: Computational complexity and assumptions on the family of warpts for the algorithms described in this survey. The computational complexity for one iteration of each algorithm is given, as a function of the number N of pixels of the template and the number n of parameters of the warps. For common applications, $N \in [10^3, 10^5]$ and $n < 10$.

References

[1] S. Baker and I. Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *CVPR*, 2001.

[2] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, pages 221–255, March 2004.

[3] S. Benhimane and E. Malis. Homography-Based 2D Visual Tracking and Servoing. *IJCV*, 2007.

[4] Helge Glöckner. Implicit functions from topological vector spaces to banach spaces. *Israel Journal of Mathematics*, 155(1):205–252, 2006.

[5] G.D. Hager and P.N. Belhumeur. Efficient Region Tracking with Parametric Models of Geometry and Illumination. *PAMI*, 20(10):1025–1039, 1998.

[6] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, pages 674–679, 1981.

[7] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *ICCV*, 2011.

[8] Heung-Yeung Shum and Richard Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.

Algo	**Objective Function** $F(\delta\mathbf{p})$	**Approx. order**	**J**	**Update**
FA	$\sum_{\mathbf{x}}(I(\mathbf{W}(\mathbf{x},\mathbf{p}_c+\delta\mathbf{p}))-T(\mathbf{x}))^2$	I	$\mathbf{J}_{FA}=\nabla I(\mathbf{W}(\mathbf{x},\mathbf{p}_c))\frac{\partial\mathbf{W}(\mathbf{x},\mathbf{p}_c)}{\partial\mathbf{p}}$	$\mathbf{p}_{c+1}=\mathbf{p}_c+\delta\mathbf{p}$
FC	$\sum_{\mathbf{x}}(I(\mathbf{W}(\mathbf{W}(\mathbf{x},\delta\mathbf{p}),\mathbf{p}_c))-T(\mathbf{x}))^2$	I	$\mathbf{J}_{FC}=\nabla I(\mathbf{W}(\mathbf{x},\mathbf{p}_c))\frac{\partial\mathbf{W}(\mathbf{x},\mathbf{p}_c)}{\partial\mathbf{x}}\frac{\partial\mathbf{W}(\mathbf{x},\mathbf{0})}{\partial\mathbf{p}}$	$\mathbf{W}(\mathbf{x},\mathbf{p}_{c+1})=\mathbf{W}(\mathbf{W}(\mathbf{x},\delta\mathbf{p}),\mathbf{p}_c)$
IC	$\sum_{\mathbf{x}}(T(\mathbf{W}(\mathbf{x},\delta\mathbf{p}))-I(\mathbf{W}(\mathbf{x},\mathbf{p}_c)))^2$	I	$\mathbf{J}_{IC}=\nabla T(\mathbf{x})\frac{\partial\mathbf{W}(\mathbf{x},\mathbf{0})}{\partial\mathbf{p}}$	$\mathbf{W}(\mathbf{x},\mathbf{p}_{c+1})=\mathbf{W}(\mathbf{W}(\mathbf{x},\delta\mathbf{p})^{-1},\mathbf{p}_c)$.
IA	$\sum_{\mathbf{x}}(I(\mathbf{W}(\mathbf{x},\mathbf{p}_c+\delta\mathbf{p}))-T(\mathbf{x}))^2$	I	$\mathbf{J}_{FA}(\mathbf{x},\mathbf{p}_c)$	$\mathbf{p}_{c+1}=\mathbf{p}_c-\delta\mathbf{p}$
ESM	$\sum_{\mathbf{x}}(I(\mathbf{W}(\mathbf{W}(\mathbf{x},\delta\mathbf{p}),\mathbf{p}_c))-T(\mathbf{x}))^2$	II	$\mathbf{J}_{ESM}=\frac{\mathbf{J}_{FC}+\mathbf{J}_{IC}}{2}$	$\mathbf{W}(\mathbf{x},\mathbf{p}_{c+1})=\mathbf{W}(\mathbf{W}(\mathbf{x},\delta\mathbf{p}),\mathbf{p}_c)$

Table 2: Formulas for the update of the warp estimate for the algorithms described in this survey. The sums are extended to all pixels $\mathbf{x}$ of the template, while $\mathbf{p}_c$ is the current parameters estimate at iteration c.