Inverse Composition for Multi-kernel Tracking

Rémi Megret, Mounia Mikram, and Yannick Berthoumieu

UMR 5131, Laboratoire d'Automatique, Productique et Signal,
Université Bordeaux 1/ENSEIRB, Talence, France
{megret, mikram, berthoum}@enseirb.fr

Abstract. Existing multi-kernel tracking methods are based on a forwards additive motion model formulation. However this approach suffers from the need to estimate an update matrix for each iteration. This paper presents a general framework that extends the existing approach and that allows to introduce a new inverse compositional formulation which shifts the computation of the update matrix to a one time initialisation step. The proposed approach thus reduces the computational complexity of each iteration, compared to the existing forwards approach. The approaches are compared both in terms of algorithmic complexity and quality of the estimation.

1 Introduction

Tracking based on color distributions [1,2,3,4] has drawn increasing interest recently, as it offers a flexible and generic framework for object tracking in videos. It is especially useful for non rigid objects, for which the integration of the information over spatially extended regions offers more allowance to slight misalignment compared to pixel based template matching [5].

The basic kernel tracking method associates a single color distribution to an object, and maximises the color similarity with a reference model using Mean-Shift [1]. Alternative approaches such as particle filters may be involved in order to take into account tracking ambiguities [6].

The parameter estimation of higher order motion models such as affine or homographic motion involves the use of an extended representation, which incorporates more information than just color. This was experimented with spatial-color distribution [7] [8] as well as with multi-kernel color distributions, where each spatial kernel is associated to a distinct color model [3,4]. An intermediate approach was recently proposed in [9] where a "texture of blobs" is used that is constituted of many overlapping kernels covering the surface of the object to track.

One of the difficulty with such multi-kernel tracking is the increased complexity introduced by the additional parameters to estimate. Previous works adopted an iterative optimisation framework based on Gauss-Newton optimisation [3,4], on quasi-Newton optimisation [9], or on the trust-region approach [10]. These methods are all based on an additive approach, where the motion parameters are incrementally refined by adding a correcting parameter until convergence.

© Springer-Verlag Berlin Heidelberg 2006

The goal of this paper is to propose a new framework for multi-kernel tracking, similar to the approach Baker and Matthews [11] introduced in image template based tracking. It covers the existing multi-kernel tracking approach, while allowing to derive a new efficient technique : the inverse compositional approach allows to use a fixed Jacobian for gradient based optimisation, which shifts a computationally costly part of the algorithm to the initialisation step and decreases the complexity of online computations.

The rest of the paper is organised as follows. In section 2, multi-kernel color distribution tracking will be exposed and the compositional framework introduced. This will serve as a global framework to present, in section 3, the classical forwards additive approach and, in section 4, the proposed inverse compositional approach. Finally, the techniques will be experimented and compared in section 5.

2 Tracking Using Color Distributions

This section presents the compositional framework for multi-kernel color distribution tracking. After formalising the notion of a multi-kernel color distribution, the framework will be exposed, and be shown to cover both the existing forwards additive approach, and a new inverse compositional approach.

2.1 Motion Model

The tracking occurs between two images I_{ref} and I related by an unknown 2D transformation $\mathbf{f}$ of parameter $\boldsymbol{\theta}^*$,

$$\forall \mathbf{x} \in \mathcal{D} \quad I_{\text{ref}}(\mathbf{x}) = I(\mathbf{f}^{-1}(\mathbf{x}, \boldsymbol{\theta}^*)) \tag{1}$$

where $\mathcal{D}$ represents a region of interest in image I_{ref}.

In the sequel, the motion model is assumed to exhibit a group property. This is the case for most models of interest, and in particular non degenerate homographies or affine motion [11]. The latter will be exploited in this paper. For the sake of notational convenience, the group property is extended to the parameters using the following notations:

$$\mathbf{f}(\,\cdot\,, \boldsymbol{\theta}^{-1}) = \mathbf{f}^{-1}(\,\cdot\,, \boldsymbol{\theta}) \tag{2}$$

$$\mathbf{f}(\,\cdot\,, \Delta\boldsymbol{\theta} \circ \boldsymbol{\theta}) = \mathbf{f}(\,\cdot\,, \Delta\boldsymbol{\theta}) \circ \mathbf{f}(\,\cdot\,, \boldsymbol{\theta}) = \mathbf{f}(\mathbf{f}(\,\cdot\,, \boldsymbol{\theta}), \Delta\boldsymbol{\theta}) \tag{3}$$

and $\boldsymbol{\theta} = 0$ represents the parameters of the identity transformation.

2.2 Multi-Kernel Color Distribution

Color distribution tracking is based on the computation of the color distribution of an image region. In the sequel, this region is defined using real valued kernels, that associate a weight to each pixel. In order to estimate non-translational movements, a general multi-kernel approach is used, that is now presented. The reader interested in a discussion on the choice of the kernels is refered to [3].

Given

- a set of κ spatial kernels $(K_1, \dots K_\kappa)$ defined as piecewise differentiable weighting functions $K_k(\mathbf{x})$ expressed in the reference coordinates $\mathbf{x}$,
- a parametric motion model $\mathbf{f}(\,\cdot\,, \boldsymbol{\theta})$ with parameter vector $\boldsymbol{\theta}$, which transforms each point $\mathbf{m}$ in the current image coordinates into a point $\mathbf{x} = \mathbf{f}(\mathbf{m}, \boldsymbol{\theta})$ in the reference coordinates, and its inverse transformation $\mathbf{m} = \mathbf{f}^{-1}(\mathbf{x}, \boldsymbol{\theta})$,
- a quantification indicative function $\delta_u(\,\cdot\,)$ whose value is 1 for colors belonging to color bin u and null otherwise,

the *multi-kernel color distribution* of image I with parameters $\boldsymbol{\theta}$ is defined as a vector $\mathbf{q}(I, \boldsymbol{\theta}) = (q_{k,u}(I, \boldsymbol{\theta}))_{k,u}$, where

$$q_{k,u}(I, \boldsymbol{\theta}) = C_k \iint_{\mathbf{x} \in \mathbb{R}^2} K_k(\mathbf{x})\, \delta_u(I(\mathbf{f}^{-1}(\mathbf{x}, \boldsymbol{\theta})))d\mathbf{x} \tag{4}$$

The normalisation constant C_k is chosen such that $\sum_u q_{k,u}(I, \boldsymbol{\theta}) = 1$ for all k.

The subvector $(q_{u,k}(I, \boldsymbol{\theta}))_u$ represents the local color distribution, over the spatial kernel K_k, of image I after it has been aligned onto the reference coordinates according to parameters $\boldsymbol{\theta}$. The relationship of (4) with expressions used in previous works [3,4,9] will be discussed in section 4.2. The choice of this expression is motivated by its invariance with respect to any 2D motion model group, even non-affine ones. Indeed

$$\mathbf{q}(I, \boldsymbol{\theta}) = \mathbf{q}(I(\mathbf{f}(\cdot, \boldsymbol{\theta}^{-1})), 0) \tag{5}$$

or more generally

$$\mathbf{q}(I, \Delta\boldsymbol{\theta} \circ \boldsymbol{\theta}) = \mathbf{q}(I(\mathbf{f}(\cdot, \boldsymbol{\theta}^{-1})), \Delta\boldsymbol{\theta}) \tag{6}$$

2.3 Compositional Framework for Multi-Kernel Tracking

Let us now consider two images related by equation (1). In the following, $\mathbf{p}$ will be used for the reference image I_{ref} and $\mathbf{q}$ for the current image I

$$\mathbf{p}(\boldsymbol{\theta}) = \mathbf{q}(I_{\text{ref}}, \boldsymbol{\theta}) \qquad \text{and} \qquad \mathbf{q}(\boldsymbol{\theta}) = \mathbf{q}(I, \boldsymbol{\theta}) \tag{7}$$

Because of (6) the following holds for any $\boldsymbol{\theta}_p$:

$$\mathbf{p}\big(\boldsymbol{\theta}_p\big) = \mathbf{q}\big(\boldsymbol{\theta}_p \circ \boldsymbol{\theta}^*\big) \tag{8}$$

Multi-kernel image alignment can be formalised as finding $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_q$ that minimise the dissimilarity between $\mathbf{p}(\boldsymbol{\theta}_p)$ and $\mathbf{q}(\boldsymbol{\theta}_q)$.

The actual matching relies on the minimisation of an error measure $E(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p)$. Several error functions can be used, such as the Bhattacharyya distance or the Kullback-Leibler divergence. Following [4] and [3], the Matusita's metric will be used in this work:

$$E(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p) = \sum_{k,u} e_{k,u}(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p)^2 \tag{9}$$

with a bin specific error vector $\mathbf{e}_{k,u}(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p)$

$$e_{k,u}(\boldsymbol{\theta}_q, \boldsymbol{\theta}_p) = \sqrt{q_{k,u}(\boldsymbol{\theta}_q)} - \sqrt{p_{k,u}(\boldsymbol{\theta}_p)} \tag{10}$$

By equating $\boldsymbol{\theta}_q$ and $\boldsymbol{\theta}_p \circ \boldsymbol{\theta}^*$ in equation 8, the estimated alignment parameter is then

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_p^{-1} \circ \boldsymbol{\theta}_q \tag{11}$$

This formalisation shows the central role that composition plays for the image alignment problem using multi-kernel distributions. We call it the compositional framework, as the effective parameter estimation $\boldsymbol{\theta}^*$ is obtained by composing the estimates $\boldsymbol{\theta}_q$ and $\boldsymbol{\theta}_p$.

This framework covers existing *forwards additive* multi-kernel tracking methods [3,4,9], which optimise the error criterion with respect to $\boldsymbol{\theta}_q = \hat{\boldsymbol{\theta}} + \Delta\boldsymbol{\theta}$. We propose to optimise with respect to $\boldsymbol{\theta}_p = \Delta\boldsymbol{\theta}$ instead, which leads to *inverse compositional* multi-kernel tracking.

The terms forward additive and inverse compositional come from the analogy with the classification Baker and Matthews [11] proposed in the context of image template tracking. The framework we introduced formalises the adaptation of this classification to the context of multi-kernel histogram based representation. In particular, a more complex error function has to be taken into account, which is a distance between histograms derived from the images, instead of direct pixelwise compensated image difference. This will play a role in the gradient based optimisation.

The forwards additive approach will be briefly presented in section 3, in order to compare its structure to the proposed inverse compositional approach, which will be presented in section 4.

3 Forwards Additive Optimisation

The forwards additive approach used in [3] relies on the Gauss-Newton optimisation of the error $E(\hat{\boldsymbol{\theta}} + \Delta\boldsymbol{\theta}, 0)$ with respect to $\Delta\boldsymbol{\theta}$, where a single iteration is estimated using

$$\Delta\boldsymbol{\theta} = A(\hat{\boldsymbol{\theta}})\ \mathbf{e}(\hat{\boldsymbol{\theta}}, 0) \tag{12}$$

where $A(\hat{\boldsymbol{\theta}})$ is an update matrix

$$A(\hat{\boldsymbol{\theta}}) = -\left(\mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}}{}^t \mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}}\right)^{-1} \mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}}{}^t \tag{13}$$

and $\mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}}$ represents the Jacobian of the error vector $\mathbf{e}(\boldsymbol{\theta}, 0)$ with respect to $\boldsymbol{\theta}$, computed at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. It can be expressed by using the gradient of $e_{u,k}$ with respect to $\boldsymbol{\theta}$, as well as the partial derivative of $\mathbf{e}$ according to each coefficient θ_m of $\boldsymbol{\theta}$

$$\mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}} = \begin{bmatrix} \vdots \\ \nabla_{e_{u,k}(\boldsymbol{\theta})|\hat{\boldsymbol{\theta}}} \\ \vdots \end{bmatrix} = \left[\cdots \left.\frac{\partial \mathbf{e}(\boldsymbol{\theta})}{\partial \theta_m}\right|_{\hat{\boldsymbol{\theta}}} \cdots \right] \tag{14}$$

A robust estimator version of this approach is used in [4]. In [9], a quasi-Newton optimisation was used instead, based on $J_{E|\hat{\boldsymbol{\theta}}}$ which also depends on $\hat{\boldsymbol{\theta}}$.

The parameter update follows the forwards additive scheme of (15), and the whole process is repeated until convergence

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \Delta\boldsymbol{\theta} \tag{15}$$

The expression of $J_{\mathbf{e}|\hat{\boldsymbol{\theta}}}$ depends on the error metric used. In particular, when using the Matusita's objective function of equation (10),

$$\mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}} = \frac{1}{2}\mathrm{diag}(\mathbf{q}(\hat{\boldsymbol{\theta}}))^{-1/2}\mathbf{J}_{\mathbf{q}|\hat{\boldsymbol{\theta}}} \tag{16}$$

where $\mathbf{J}_{\mathbf{q}|\boldsymbol{\theta}}$, the Jacobian of $\mathbf{q}(\boldsymbol{\theta})$, will be studied in more details in section 4.2.

4 Inverse Compositional Optimisation

In the previous approach, the computation of equations (12) and (13) is the bottleneck of the algorithm. Indeed, the update matrix $A(\hat{\boldsymbol{\theta}})$ needs to be computed for each new iteration, which involves in particular the computation of $\mathbf{J}_{e|\boldsymbol{\theta}}{}^{t}\mathbf{J}_{e|\boldsymbol{\theta}}$.

An alternative approach is now proposed, that takes advantage of the general framework introduced in section 2.3 and allows to use a constant update matrix A, which can be pre-computed once during the model initialisation.

4.1 Principle

In a similar way as the forwards approach, a Gauss-Newton iteration is computed, but the parameter correction now applies to the kernel position in the reference image. The Gauss-Newton parameter update of $E(\hat{\boldsymbol{\theta}}, \Delta\boldsymbol{\theta})$ with respect to $\Delta\boldsymbol{\theta}$ satisfies:

$$\Delta\boldsymbol{\theta} = A\ \mathbf{e}(\hat{\boldsymbol{\theta}}, 0) \tag{17}$$

The update matrix A is now a constant matrix

$$A = -\left(\mathbf{J}_{\hat{\mathbf{e}}|0}{}^{t}\mathbf{J}_{\hat{\mathbf{e}}|0}\right)^{-1}\mathbf{J}_{\hat{\mathbf{e}}|0}{}^{t} \tag{18}$$

where $\mathbf{J}_{\hat{\mathbf{e}}|0}$ represents the Jacobian of $\mathbf{e}(\hat{\boldsymbol{\theta}}, \Delta\boldsymbol{\theta})$ with respect to $\Delta\boldsymbol{\theta}$ computed at $\Delta\boldsymbol{\theta} = 0$. When using the Matusita metric, $\mathbf{J}_{\hat{\mathbf{e}}|0}$ does not depend on $\hat{\boldsymbol{\theta}}$.

$$\mathbf{J}_{\hat{\mathbf{e}}|0} = -\ \frac{1}{2}\mathrm{diag}(\mathbf{p}(0))^{-1/2}\mathbf{J}_{\mathbf{p}|0} \tag{19}$$

The estimation rule (17) can be compared to the analogous rule (12) in the forwards approach, as it has the same structure. The main difference is that in the inverse approach, the A matrix does not depend on $\hat{\boldsymbol{\theta}}$ anymore, which allows to pre-compute it offline, thus removing most of the online complexity.

In this approach, the correction parameter $\Delta\boldsymbol{\theta}$ represents an update on the kernel locations in the reference image. In order to convert it into an updated parameter vector for the motion between the two images, the compositional framework is invoked through equation (11), which corresponds to the update rule:

$$\hat{\boldsymbol{\theta}} \leftarrow \Delta\boldsymbol{\theta}^{-1} \circ \hat{\boldsymbol{\theta}} \tag{20}$$

The estimate $\hat{\boldsymbol{\theta}}$ is iteratively updated while it corresponds to a decrease of the error $E(\hat{\boldsymbol{\theta}}, 0)$ and until convergence.

4.2 Jacobian Computation

The computation of the Jacobian $\mathbf{J}_{\mathbf{p},0}$ or its more general form $\mathbf{J}_{\mathbf{q},\boldsymbol{\theta}}$ is not direct from equation (4), as δ_u is not easily differentiable. This part is detailed in the current section through the computation of the gradient of $q_{k,u}$.

An equivalent formulation of $q_{k,u}$ is used in [9], which is based on the coordinates $\mathbf{m}$ in the current image :

$$q_{k,u}(I, \boldsymbol{\theta}) = C_k \iint_{\mathbf{m}} K_k(\mathbf{f}(\mathbf{m}, \boldsymbol{\theta}))\ \delta_u(I(\mathbf{m}))\ j(\mathbf{m}, \boldsymbol{\theta})\ d\mathbf{m} \tag{21}$$

where $j(\mathbf{m}, \boldsymbol{\theta}) = \left|\mathbf{J}_{\mathbf{f}(\mathbf{m},\boldsymbol{\theta})|\mathbf{m}}\right|$ is the absolute value of the determinant of the Jacobian of $\mathbf{f}$ with respect to $\mathbf{m}$.

For affine transformations, $j(\mathbf{m}, \boldsymbol{\theta})$ is constant with respect to $\mathbf{m}$, which leads to a simplified expression

$$q_{k,u}(I, \boldsymbol{\theta}) = C_{k,\boldsymbol{\theta}} \iint_{\mathbf{m}} K_k(\mathbf{f}(\mathbf{m}, \boldsymbol{\theta}))\ \delta_u(I(\mathbf{m}))\ d\mathbf{m} \tag{22}$$

with $C_{k,\boldsymbol{\theta}} = C_k j(\mathbf{m}, \boldsymbol{\theta})$ corresponding to the kernel normalisation parameter that now depends on $\boldsymbol{\theta}$. This equation is very similar to the definitions of $q_{k,u}$ used in [4,3]. Note that for non-affine motion this equivalence does not hold, so that the computation of the Jacobian for more complex models should instead use the full expression (21).

By differentiating (22) and after taking into account the kernel normalisation $\sum_u q_{k,u} = 1$, the gradient can be simplified as (23).

$$\nabla_{q_{u,k}|\hat{\boldsymbol{\theta}}} = C_{k,\hat{\boldsymbol{\theta}}} \iint_{\mathbf{m}} \nabla_{K_k(\mathbf{x})|\mathbf{f}(\mathbf{m},\hat{\boldsymbol{\theta}})}\, \mathbf{J}_{\mathbf{f}(\mathbf{m},\boldsymbol{\theta})|\hat{\boldsymbol{\theta}}} \left(\delta_u(I(\mathbf{m})) - q_{k,u}(\hat{\boldsymbol{\theta}})\right) d\mathbf{m} \tag{23}$$

In the simpler case of inverse composition, the Jacobian is

$$\nabla_{p_{u,k}|0} = C_k \iint_{\mathbf{x}} \nabla_{K_k(\mathbf{x})|\mathbf{x}}\, \mathbf{J}_{\mathbf{f}(\mathbf{x},\Delta\boldsymbol{\theta})|0} \left(\delta_u(I_{\text{ref}}(\mathbf{x})) - p_{k,u}(0)\right) d\mathbf{x} \tag{24}$$

The $-q_{k,u}(\hat{\boldsymbol{\theta}})$ term in the previous equations is related to the gradient of the normalisation constant $C_{k,\boldsymbol{\theta}}$ with respect to $\boldsymbol{\theta}$. Its influence is null for translation and rotation components, but should be taken into account when considering scale. Neglecting this term, as was done in [4], results in a biased estimation of the Jacobian. This is illustrated for the scale estimation in figure 3(d), where it leads to the under-estimation of the motion parameters.

From a practical point of view, the integrals must be replaced by discrete sums, on either the integer pixels $\{\mathbf{m}_i\}$ of the current image for (23) and (24), or on a regular grid $\{\mathbf{x}_j\}$ in the reference coordinates for (23). In order to improve the running time, this sampling was done on $\mathbf{x}$, which allows to pre-compute the kernel values and gradients at the sampling points.

A large choice of kernel functions can be used in this framework in the same way as with the forwards approach [3]. In this paper, Epanechnikov kernels are used. The kernel K_k, with centre $\mathbf{x}_k$ and covariance matrix B_k is defined by

$$K_k(\mathbf{x}) = \max(0, 1 - (\mathbf{x} - \mathbf{x}_k)^t {B_k}^{-1} (\mathbf{x} - \mathbf{x}_k)) \tag{25}$$

$$\nabla_{K_k(\mathbf{x})|\mathbf{x}} = -2(\mathbf{x} - \mathbf{x}_k)^t {B_k}^{-1} \quad \text{where } K_k(\mathbf{x}) > 0 \tag{26}$$

5 Experiments and Discussion

The properties of the forwards additive multi-kernel tracking approach have been experimentally studied and compared to the image template based approach in [4]. It was shown that the kernel approach allows for a larger region of convergence, thanks to the integration of the kernels. This comes at the cost of a slightly less precise alignment, which was resolved by combining the two approaches. As the inverse compositional approach proposed in this paper uses the same multi-kernel representation as the forwards approach, these experiments will not be duplicated here. The proposed method is expected to be faster to compute than the forwards method because of its algorithmic structure, while bearing similar tracking performances. This section will therefore be devoted to checking this hypothesis.

5.1 Computational Performance

The algorithmic structures of both the forwards additive and the inverse compositional are summed up and compared in figure 1.

For the complexity analysis, the following notations will be used : κ is the number of kernels, P the mean number of pixels for which a kernel is non null, U the number of color bins in each color histogram and T the number of degrees of freedom in $\boldsymbol{\theta}$. The cost of each step is of the order of $\mathcal{O}(\kappa P)$ for $\{1\}$ and $\{3\}$, $\mathcal{O}(\kappa U)$ for $\{4\}$, $\mathcal{O}(\kappa PT)$ for $\{5\}$, $\mathcal{O}(\kappa UT^2)$ for the computation of ${\mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}}}^t \mathbf{J}_{\mathbf{e}|\hat{\boldsymbol{\theta}}}$ and $\mathcal{O}(T^3)$ for its inversion in $\{6\}$, and $\mathcal{O}(\kappa UT)$ for $\{7\}$.

Given that U and P are large compared to the other parameters (of the order of 100 to 1000), steps $\{5\}$ and $\{6\}$ are the two most costly steps in the algorithm. Therefore, moving them to a pre-computation phase decreases the overall complexity of each iteration significantly.

In particular, with our current Matlab implementation, one iteration for $\kappa = 9$ kernels each covering $P = 150$ pixels and with the color quantised into $U = 64$

Forwards Additive		Inverse Compositional	
Pre-computations		Pre-computations	
{1} Reference distribution $\mathbf{p}$	(7)	**{1}** Reference distribution $\mathbf{p}$	(7)
		{5} Jacobian $\mathbf{J}_{\mathbf{e}\vert 0}$	(19) (24)
		{6} Update matrix A	(18)
For each new frame		For each new frame	
{2} Initial estimate $\hat{\boldsymbol{\theta}}$		**{2}** Initial estimate $\hat{\boldsymbol{\theta}}$	
Iterate until convergence:		Iterate until convergence:	
{3} Current distribution $\mathbf{q}(\hat{\boldsymbol{\theta}})$	(7)	**{3}** Current distribution $\mathbf{q}(\hat{\boldsymbol{\theta}})$	(7)
{4} Current error $\mathbf{e}(\hat{\boldsymbol{\theta}}, 0)$	(10)	**{4}** Current error $\mathbf{e}(\hat{\boldsymbol{\theta}}, 0)$	(10)
{5} Jacobian $\mathbf{J}_{\mathbf{e}\vert\hat{\boldsymbol{\theta}}}$	(16) (23)		
{6} Update matrix $A(\hat{\boldsymbol{\theta}})$	(13)		
{7} Step $\Delta\boldsymbol{\theta}$	(12)	**{7}** Step $\Delta\boldsymbol{\theta}$	(17)
{8} New estimate : $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \Delta\boldsymbol{\theta}$		**{8}** New estimate : $\hat{\boldsymbol{\theta}} \leftarrow \Delta\boldsymbol{\theta}^{-1} \circ \hat{\boldsymbol{\theta}}$	

Fig. 1. Algorithm comparison. The pre-computations occur only during the model initialisation, and is not repeated for a new frame. For each step, the equation that defines the related computation is shown at the right.

color bins with an affine motion model ($T = 6$) requires 168 ms with the inverse compositional approach, instead of 359 ms with the classical forwards additive approach.

5.2 Convergence Properties

The forwards additive approach is a typical Gauss-Newton optimisation of the error $E(\hat{\boldsymbol{\theta}}, 0)$. The inverse compositional approach adopts an hybrid scheme. Indeed, the general optimisation criterion is still $E(\hat{\boldsymbol{\theta}}, 0)$, but each iteration uses the $E(\hat{\boldsymbol{\theta}}, \Delta\boldsymbol{\theta})$ criterion. These functions both express the matching error as was shown in section 2.3. They are not identical when the error is large, which is why the convergence properties of the two approaches are now compared.

Figure 3(a-d) shows the return map of the two methods (forwards additive from section 3, inverse compositional from section 4), when perturbated with a pure translation (b), a pure rotation (c), and a pure scale (d). Nine Epanechnikov kernels centred on a regular 3×3 grid were used, as shown[1] in (a).

Overall, both approaches yield similar results. Indeed, both the forwards and the inverse methods approximate well the correction for small perturbations, and tend to under-estimate the correction for larger perturbations. This observation reflect the fact that both are based on a linearisation of the error around the initial parameters, which is valid for small perturbations.

[1] Test image courtesy of Krystian Mikolajczyk, `http://www.robots.ox.ac.uk/~vgg`.

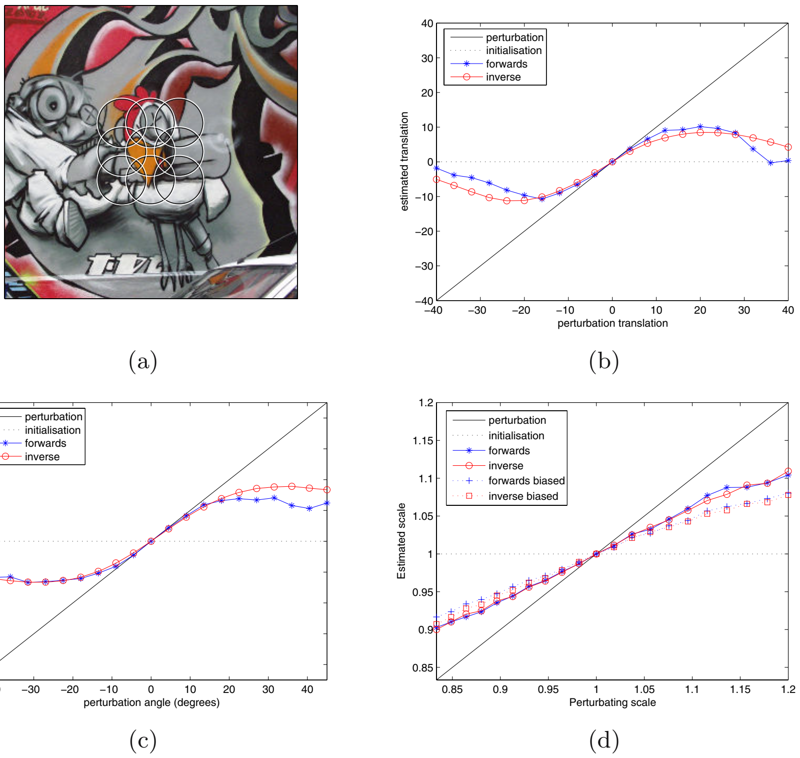
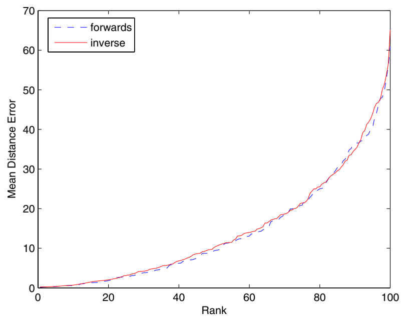


Fig. 2. Comparison of the parameter estimation of a single iteration for controled affine perturbations (see section 5.2). (a) Test image, with the supports of the nine Epanechnikov kernels overlayed. Return maps (corrected parameters depending on the perturbated parameters: the closer to the perturbation the better) for (b) an horizontal shift, (c) a rotation around the centre of the image.

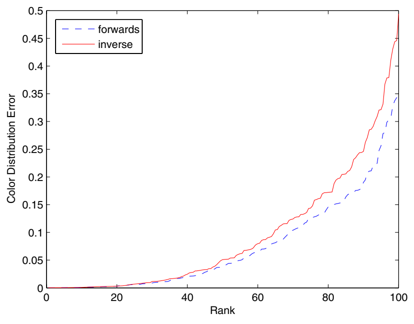
The scale estimation exhibits a systematic under-estimation on this example. This is observed with other classical kernel configurations, but not when using a totally unambiguous image made of squares with unique colors. Although the unbiased approach presented in section 4.2 slightly improves the estimation, further work is needed to explain this behaviour. The estimations are nevertheless in the correct direction even for large perturbations, which make the iterative optimisation eventually converge to the correct parameters even in that case.

The quality of the parameter estimation is also evaluated in more general conditions in figure 3(e-f), for one single iteration. The perturbations are a combination of random translations within $[-20, 20]$ pixels, rotations within $[-20, 20]$ degrees and scale within $[1.2^{-1}, 1.2]$.

The mean spatial error D corresponds to the average of the spatial error of the kernel centers, evaluated in the reference coordinates. These two measures allow to evaluate translation, rotation and scale errors in an unified manner.

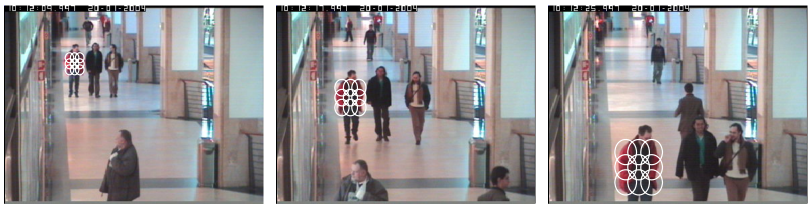


(e)

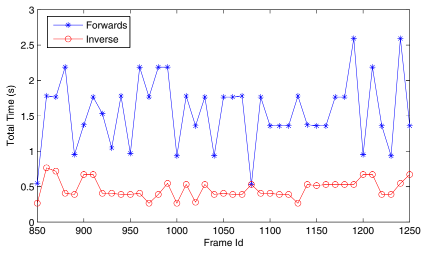


(f)

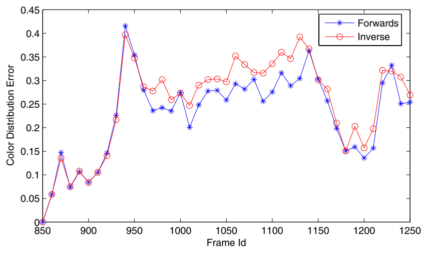
Fig. 3. Comparison of the parameter estimation quality of a single iteration for random affine perturbations (see section 5.2). Ranked mean spatial error (e) and color distribution error (f).



850 1050 (a) 1250



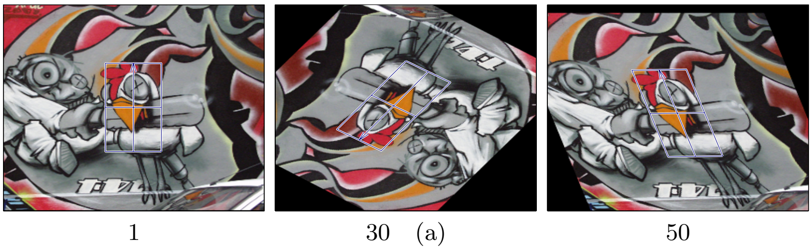
(b)

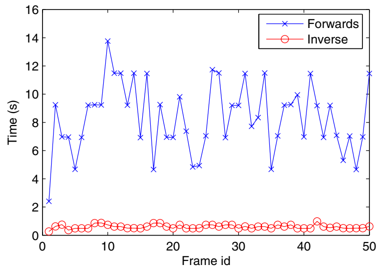


(c)

Fig. 4. Tracking example with scale change on CAVIAR video for the inverse compositional approach (a). Comparison of the computational time (b) and the color distribution error $E(\boldsymbol{\theta}, 0)$ after convergence (c) for each frame on the same video, with both approaches.

Results are sorted by increasing error. These results show that the inverse approach has a slightly larger color distribution error than the forwards approach (f), which can be explained by the fact that it does not operate directly on the optimisation criterion $E(\hat{\boldsymbol{\theta}}, 0)$. This difference do not seem to impact the parameter estimation, though, as the corrected parameter appear to be equally good from a spatial point of view (e).





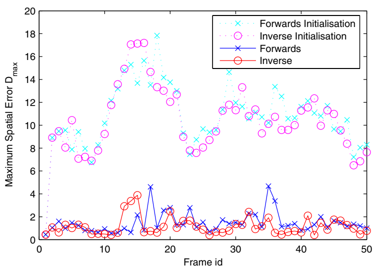


Fig. 5. Tracking example for a video with affine distortions, with the parameter estimation overlayed for the inverse approach (a). Comparison of the computational time (b). Comparison of the mean spatial error (c); the error for the initialisation at each frame is plotted to show the amount of correction needed on this sequence.

5.3 Tracking

In this section the computational performance and the quality of estimation are compared in tracking conditions: a person with scale change[2] in figure 4 and an image part with affine distortions in figure 5. Both trackings use 9 Epanechnikov kernels centred on a regular 3×3 grid. The parameters obtained with the forwards and the inverse approach are very similar, which is supported by similar mean spatial errors D with respect to the ground truth in figure 5-c, and comparable color distribution errors E in figure 5-c. In the last case, a slightly lower error is observed for the forwards approach, which was discussed in section 5.2. The computational time is in both cases significantly reduced by using the inverse approach instead of the forwards approach.

6 Conclusion

This paper presented the adaptation and the application of inverse composition, which is already used in image template tracking, to tracking with multi-kernel color distributions. The multi-kernel tracking paradigm was reformalised in order to cover both the existing forwards additive approach and a new inverse

[2] Video produced by the EC Funded CAVIAR project/IST 2001 37540. `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`

compositional approach. The quality of the parameter estimation of the new technique is similar to the multi-kernel forwards additive approach, while shifting the computational burden from each iteration to a one-time initialisation step.

The structure of the proposed approach relies on an iterative optimisation with a constant update matrix A, which is estimated by inverting the Jacobian of the error function. This structure offers the possibility to introduce alternative forms for A, such as the hyperplane approximation [12], in a multi-kernel context.

Other interesting problems for future research would be to study how illumination changes, which can be taken into account in the forwards approach [4,9], could be handled in an inverse compositional approach, and how the choice of the kernel configuration impacts the performances of the method.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 564–575
2. Fan, Z., Wu, Y., Yang, M.: Multiple collaborative kernel tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA (2005) 502–509
3. Hager, G.D., Dewan, M., Stewart, C.V.: Multiple kernel tracking with SSD. In: IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA (2004) 790–797
4. Georgescu, B., Meer, P.: Point matching under large image deformations and illumination changes. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 674–688
5. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision **56** (2004) 221–255
6. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: European Conference on Computer Vision, ECCV'2002, LNCS 2350, Copenhaguen, Denmark (2002) 661–675
7. Elgammal, A.M., Duraiswami, R., Davis, L.S.: Probabilistic tracking in joint feature-spatial spaces. In: IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI, USA (2003) 781–788
8. Zhang, H., Huang, W., Huang, Z., Li, L.: Affine object tracking with kernel-based spatial-color representation. In: IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA (2005) 293–300
9. Guskov, I.: Kernel-based template alignment. In: IEEE Conference on Computer Vision and Pattern Recognition, New-York, USA (2006) 610–617
10. Liu, T.L., Chen, H.T.: Real-time tracking using trust-region methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2003) 397–402
11. Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA (2001) 1090–1097
12. Jurie, F., Dhome, M.: Hyperplane approximation for template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 996–1000