Appeared in the Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition

Equivalence and Efficiency of Image Alignment Algorithms

Simon Baker and Iain Matthews

The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213

Abstract

There are two major formulations of image alignment using gradient descent. The first estimates an additive increment to the parameters (the *additive* approach), the second an incremental warp (the *compositional* approach). We first prove that these two formulations are equivalent. A very efficient algorithm was recently proposed by Hager and Belhumeur using the additive approach that unfortunately can only be applied to a very restricted class of warps. We show that using the compositional approach an equally efficient algorithm (the *inverse compositional* algorithm) can be derived that can be applied to any set of warps which form a *group*. While most warps used in computer vision form groups, there are a certain warps that do not. Perhaps most notable is the set of piecewise affine warps used in Flexible Appearance Models (FAMs). We end this paper by extending the inverse compositional algorithm to apply to FAMs.

1 Introduction

Image *alignment* or *registration* consists of moving, and possibly deforming, a template to minimize the difference between the template and an image. Some of the applications of alignment include optical flow [Lucas and Kanade, 1981], tracking [Black and Jepson, 1998, Hager and Belhumeur, 1998, Cascia *et al.*, 2000], parametric and layered motion estimation [Bergen *et al.*, 1992], mosaic-ing [Shum and Szeliski, 2000], and face coding [Cootes *et al.*, 1998].

The usual approach to image alignment is gradient descent. Various other numerical algorithms (such as *difference decomposition* [Gleicher, 1997, Cascia *et al.*, 2000]) have also been proposed, but gradient descent is the defacto standard. There are several different formulations of gradient descent, however. One major difference between the various algorithms is whether they estimate an additive increment to the parameters [Lucas and Kanade, 1981] (an approach which we will call *additive*), or whether instead they estimate an incremental warp [Shum and Szeliski, 2000] (an approach which we will refer to as *compositional*.)

The first part of this paper proves that these two approaches are equivalent in the sense that they take the same steps in each iteration (to a first order approximation.) One difference between the two formulations, however, is that additive algorithms can be applied to any type of warp, whereas compositional algorithms can only be applied to sets of warps that form *semi-groups*. The incremental warp must be *composed* with the current estimate of the warp and so the set of warps must be closed under composition.

Another difference between the various algorithms is their efficiency. For example, [Hager and Belhumeur, 1998] recently proposed a very efficient algorithm. The key step in the derivation of their algorithm is to apply a change of variables to *invert* the role of the image and the template. To do this the Jacobian of the change of variables must take a particularly simple form. As a result their algorithm can unfortunately only be used with translations, 2D similarity transforms, affine warps, and certain other esoteric warps.

Hager and Belhumeur use the additive formulation. We therefore call their algorithm the *inverse additive* algorithm. A natural question then, is what happens if we apply the same change of variables in the compositional formulation? It turns out that the change of variables in this case is always the identity, the Jacobian of which is 1, to a first order approximation. Noticing this fact immediately leads us to a new efficient image alignment algorithm that can be applied to much wider class of warps. The change of variables means that every warp in the set must now be invertible, but that is the only new restriction. The *inverse compositional* algorithm proposed in this paper can be applied to any set of warps that form a *group*. This includes many warps that the inverse additive algorithm cannot be applied to, such as homographies and 3D rotations [Shum and Szeliski, 2000].

Although nearly all warps used in computer vision are groups, there is one important set that is not, the piecewise affine warps used in Flexible Appearance Models[1] (FAMs), Active Appearance Models (AAMs) [Cootes *et al.*, 1998], and Active Blobs [Sclaroff and Isidoro, 1998]. Although the inverse compositional algorithm cannot be *directly* used with piecewise affine warps, in the final part of this paper we show how it can be extended to apply to such warps. The approach is to derive first order approximations to the inversion and composition operators. Until now, the users of piecewise affine warps have had to resort to "non gradient descent" algorithms in order to obtain efficiency. Our image alignment framework leads naturally to the first effi-

[1] We use the term *Flexible Appearance Model* for models based on piecewise affine warps and which have independent shape and appearance eigenspaces, unlike AAMs which have coupled eigen-spaces.

cient gradient descent algorithm for FAMs.

2 Equivalence

Suppose we are trying to align a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$, where $\mathbf{x} = (x, y)^{\mathrm{T}}$ is a vector containing the image coordinates. If the warp is denoted by $\mathbf{W}(\mathbf{x}; \mathbf{p})$, where $\mathbf{p} = (p_1, \ldots p_n)^{\mathrm{T}}$ is a vector of parameters, we assume that the goal of image alignment is to minimize:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x}) \right]^2 \tag{1}$$

with respect to $\mathbf{p}$, where the sum is performed over the pixels $\mathbf{x}$ in the template image $T(\mathbf{x})$.

2.1 Additive Image Alignment

The additive approach assumes that a current estimate of $\mathbf{p}$ is known and then iteratively solves for increments to the parameters $\Delta\mathbf{p}$; i.e. the following expression is minimized:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta\mathbf{p})) - T(\mathbf{x}) \right]^2 \tag{2}$$

with respect to $\Delta\mathbf{p}$. Performing a first order Taylor expansion on this expression gives:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 . \tag{3}$$

This is a least squares problem, the solution of which is:

$$\Delta\mathbf{p} \;=\; \sum_{\mathbf{x}} H^{-1} \left[\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right] \tag{4}$$

where H is the $n \times n$ *Hessian* matrix:

$$H \;=\; \sum_{\mathbf{x}} \left[\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] . \tag{5}$$

The additive algorithm [Lucas and Kanade, 1981] consists of iterating the following steps until the estimates of the parameters $\mathbf{p}$ converge:

1. Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$;
2. Compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$;
3. Warp the gradient of image I to compute $\boldsymbol{\nabla} I$;
4. Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$;
5. Compute the Hessian matrix using Equation (5);
6. Compute $\Delta\mathbf{p}$ using Equation (4);
7. Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$.

Because the warped gradient $\boldsymbol{\nabla} I$ and the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ both, in general, depend on $\mathbf{p}$, all of these steps must be performed in every iteration of the algorithm. The estimate of the parameters $\mathbf{p}$ varies from iteration to iteration.

2.2 Compositional Image Alignment

The compositional approach also assumes that a current estimate of $\mathbf{p}$ is known, but iteratively solves for an an incremental warp $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$ rather than an additive update to $\mathbf{p}$ [Shum and Szeliski, 2000]; i.e. the following is minimized:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \Delta\mathbf{p}); \mathbf{p})) - T(\mathbf{x}) \right]^2 \tag{6}$$

with respect to $\Delta\mathbf{p}$. A first order Taylor expansion gives:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \mathbf{0}); \mathbf{p})) + \boldsymbol{\nabla} I(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 . \tag{7}$$

where $I(\mathbf{W})(\mathbf{x})$ is the warped image $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$. Assuming (without loss of generality) that $\mathbf{W}(\mathbf{x}; \mathbf{0})$ is the identity, then $I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \mathbf{0}), \mathbf{p})) = I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$. There are then two differences between Equations (3) and (7). The first difference is that the gradient of $I(\mathbf{x})$ is replaced with the gradient of $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$. The second difference is hidden by the concise notation. The Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x}; \mathbf{p})$ in Equation (3), but in Equation (7) it is evaluated at $(\mathbf{x}; \mathbf{0})$; i.e. where the Taylor expansion was performed.

The only changes to the algorithm are therefore: (1) the gradient of $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ should be used in Step 3, (2) the Jacobian should be evaluated at $(\mathbf{x}; \mathbf{0})$ in Step 4, and (3) the warp is updated $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$ in Step 7. The Jacobian in Step 4 is a constant across iterations and can be pre-computed. (It is also generally simpler analytically [Shum and Szeliski, 2000].) On the other hand, the update of the warp is more complex. Instead of simply adding the updates $\Delta\mathbf{p}$ to the current estimate of the parameters $\mathbf{p}$, the incremental update to the warp $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$ must be *composed* with the current estimate $\mathbf{W}(\mathbf{x}; \mathbf{p})$. This operation typically involves multiplying two matrices, although for more complex warps it can be more involved. The warps must also form a *semi-group* if $\mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$ is always to be a valid warp, and $\mathbf{W}(\mathbf{x}; \mathbf{0})$ is to be the identity.

2.3 Proof of Equivalence

In the additive formulation we minimize:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 \tag{8}$$

with respect to $\Delta\mathbf{p}$ and then update $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$. The corresponding update to the warp is:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) \;\leftarrow\; \mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta\mathbf{p}) \;\approx\; \mathbf{W}(\mathbf{x}; \mathbf{p}) + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \tag{9}$$

when a first order Taylor expansion is made in $\Delta\mathbf{p}$. In the compositional formulation we minimize:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x};\mathbf{0});\mathbf{p})) + \boldsymbol{\nabla} I(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 . \tag{10}$$

where $\boldsymbol{\nabla} I(\mathbf{W})$ is the gradient of $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$, which equals $\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{x}}$ by the chain rule. Assuming that $\mathbf{W}(\mathbf{x};\mathbf{0})$ is the identity warp, Equation (10) simplifies further to:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 . \tag{11}$$

In the compositional approach, the update to the warp is $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$. In order to simplify this expression, note that:

$$\mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \;\approx\; \mathbf{W}(\mathbf{x};\mathbf{0}) + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \;=\; \mathbf{x} + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \tag{12}$$

is the first order Taylor expansion of $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ and that:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \;=\; \mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p}). \tag{13}$$

Combining these last two equations, and applying the Taylor expansion again, gives the update in the compositional formulation as:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \;\leftarrow\; \mathbf{W}(\mathbf{x};\mathbf{p}) + \frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p}. \tag{14}$$

The only difference between the additive formulation in Equations (8) and (9), and the compositional formulation in Equations (11) and (14) is that $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is replaced by $\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$. Equations (8) and (11) therefore generally result in different estimates for $\Delta\mathbf{p}$. The overall updates to the warp are the same to first order in $\Delta\mathbf{p}$, however. The warp update vectors $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ in the additive formulation and $\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ in the compositional formulation both span the same linear space, the tangent space of $\mathbf{W}(\mathbf{x};\mathbf{p})$. The optimal value of $\frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p}$ in Equation (8) will therefore equal the optimal value of $\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p}$ in Equation (11) and so the updates are equal; i.e. we have proved that the two formulations are equivalent.

2.4 Modeling Appearance Variation

Often it is assumed that $T(\mathbf{x})$ is not just a single image, but is actually a single image plus an unknown vector in a (known) linear subspace. Often the linear subspace is used to model illumination change [Hager and Belhumeur, 1998, Cascia *et al.*, 2000], but could easily model more general appearance variation [Cootes *et al.*, 1998, Black and Jepson, 1998]. We now briefly describe how either of the equivalent algorithms can be extended to allow appearance variation. (See the technical report [Baker *et al.*, 2001] for the details.)

Suppose that the images $A_1(\mathbf{x}), \ldots, A_d(\mathbf{x})$ are an orthonormal basis for the appearance linear subspace. Image alignment is then posed as minimizing:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) - \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right]^2 \tag{15}$$

simultaneously over the vector of parameters $\mathbf{p}$ and the appearance coefficients λ_i. If we denote the linear subspace by $\mathrm{span}(A_i)$ and its orthogonal complement by $\mathrm{span}(A_i)^\perp$, the expression in Equation (15) can be rewritten as:

$$\left\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) - \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right\|^2_{\mathrm{span}(A_i)^\perp} + \left\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) - \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right\|^2_{\mathrm{span}(A_i)} \tag{16}$$

where $\| \cdot \|^2_L$ denotes the square of the Euclidean norm of the vector projected into the linear subspace L. Since the norm only considers the components of vectors in the orthogonal complement of $\mathrm{span}(A_i)$, any component in $\mathrm{span}(A_i)$ itself can be dropped. We therefore have to minimize:

$$\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) \|^2_{\mathrm{span}(A_i)^\perp} + \left\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) - \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right\|^2_{\mathrm{span}(A_i)} . \tag{17}$$

The first of these two terms does not depend upon λ_i. For any $\mathbf{p}$, the minimum value of the second term is always 0. The minimization can therefore be performed *sequentially* by first minimizing the first term with respect to $\mathbf{p}$ alone, and then minimizing the second term with respect to λ_i.

Minimizing the first term in Equation (17) is not really any different to solving the original alignment problem. We just need to work in the linear subspace $\mathrm{span}(A_i)^\perp$; i.e. we project $\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ into $\mathrm{span}(A_i)^\perp$ in Equations (4) and (5). The error image does not need to be projected into this subspace because if one of the two terms in a dot product is projected into a linear subspace, the result is the same as if they both were. Minimizing the second term in Equation (17) has the closed-form solution:

$$\lambda_i \;=\; \sum_{\mathbf{x}} A_i(\mathbf{x}) \cdot \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) \right] . \tag{18}$$

The description here has been in terms of the additive formulation, but the first term in Equation (17) can alternatively be optimized with a compositional algorithm.

3 Efficiency

As a number of authors have pointed out, there is a huge computational cost in re-evaluating the Hessian in every iteration (Steps 3–5) of the algorithm [Hager and Belhumeur, 1998, Dellaert and Collins, 1999, Shum and Szeliski, 2000]. If only the Hessian were a constant, it could just be pre-computed and then re-used. Each iteration of the algorithm would then just consist of an image warp (Step 1), an image difference (Step 2), a collection of image "dot-products" (Step 6), and the update to the parameters (Step 7). All of these operations are very simple and can easily be performed at (close to) frame-rate [Dellaert and Collins, 1999].

Unfortunately the Hessian is, in general, a function of $\mathbf{p}$ in both the additive and the compositional formulations. Although various approximate solutions can be used (such as only updating the Hessian every few iterations and efficiently approximating the Hessian [Shum and Szeliski, 2000]) these approximations are all inelegant, and it is often hard to say how good approximations they really are. It would be far better if the problem could be reformulated in an equivalent way in which the Hessian is exactly constant.

3.1 Inverse Additive Image Alignment

The key to efficiency is switching the role of the image and the template, as in [Hager and Belhumeur, 1998], to yield the *inverse additive* algorithm. There, the authors change variables $\mathbf{y} = \mathbf{W}(\mathbf{x};\mathbf{p})$ or $\mathbf{x} = \mathbf{W}(\mathbf{y};\mathbf{p})^{-1}$. Because the summation in Equation (1) is a discrete approximation to an integral, the Jacobian (with respect to $\mathbf{y}$) of the warp $\mathbf{W}(\mathbf{y};\mathbf{p})^{-1}$ has to be incorporated when the change of variables is performed. Equation (1) therefore becomes:

$$\sum_{\mathbf{y}} \left| \frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}} \right| \cdot \left[I(\mathbf{y}) - T(\mathbf{W}(\mathbf{y};\mathbf{p})^{-1}) \right]^2 \qquad (19)$$

where the summation is over the sub-region of I that corresponds to the template T warped with $\mathbf{W}(\mathbf{x};\mathbf{p})$,

Much of [Hager and Belhumeur, 1998] is concerned with the Jacobian $\frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}}$. Hager and Belhumeur have to assume that this Jacobian has a special form to proceed, namely that the product of it with the other Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ (see Equation (20) in [Hager and Belhumeur, 1998]) can be factored into a component that only depends upon $\mathbf{p}$ (and which can be moved out of the summation and dealt with later), and a second component that only depends upon $\mathbf{x}$ (which becomes an iteration independent weighting factor.)

The full details of the inverse additive algorithm are outside the scope of this paper. But, it is this assumption about the product of the two Jacobians that results in the inverse additive algorithm only being applicable to a small number of warps: 2D translations, 2D similarity transforms, 2D affine warps, and a small number of more esoteric warps.

3.2 Inverse Compositional Image Alignment

The main focus of this paper is the *inverse compositional* algorithm, and its extension to FAMs. The inverse compositional algorithm is derived in a similar way to the algorithm of [Hager and Belhumeur, 1998] but uses the compositional formulation rather than the additive one. The proof of equivalence follows in the next section, but the result is that the algorithm minimizes:

$$\sum_{\mathbf{x}} \left[T(\mathbf{W}(\mathbf{x};\Delta\mathbf{p})) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2 \qquad (20)$$

with respect to $\Delta\mathbf{p}$. (Note that the roles of I and T are reversed.) Performing a first order Taylor expansion gives:

$$\sum_{\mathbf{x}} \left[T(\mathbf{W}(\mathbf{x};\mathbf{0})) + \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2 . \qquad (21)$$

Assuming again without loss of generality that $\mathbf{W}(\mathbf{x};\mathbf{0})$ is the identity, the solution to this least-squares problem is:

$$\Delta\mathbf{p} \; = \; -\sum_{\mathbf{x}} H^{-1} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right] \qquad (22)$$

where H is the Hessian matrix with I replaced by T:

$$H \; = \; \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] \qquad (23)$$

and the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x};\mathbf{0})$. Since there is nothing in the Hessian that depends upon $\mathbf{p}$, it is constant across iterations and can be pre-computed. The algorithm then becomes iterating the following four steps until the parameters $\mathbf{p}$ converge:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$;
2. Compute the error image $I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})$;
3. Compute $\Delta\mathbf{p}$ using Equation (22);
4. Update: $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$.

This algorithm is much more efficient than the forwards algorithms. Steps (3-5) of the forwards algorithms need only be performed once as a pre-computation, rather than once per iteration. The only extra cost is inverting $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ and composing it with $\mathbf{W}(\mathbf{x};\mathbf{p})$. This typically requires a matrix inversion and a matrix multiplication on small (3×3 for the homography) matrices. Potentially these two steps could be more involved as we will see in Section 4. The inverse compositional algorithm is almost exactly as efficient as the inverse additive algorithm [Hager and Belhumeur, 1998]. It can, however, be applied to any warps that form a *group*, including homographies and 3D rotations, rather than only to a small collection of warps. The group property is required to perform Step 4 of the algorithm.

Note that a restricted version of the inverse compositional algorithm was proposed (for homographies only) in

[Dellaert and Collins, 1999]. We have shown that the algorithm can be applied to a much wider class of warps. Also note that the appearance variation extension in Section 2.4 also applies to the inverse compositional algorithm [Baker *et al.*, 2001]. The only change needed to the algorithm is projecting $\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ into $\mathrm{span}(A_i)^{\perp}$ in Equations (22–23).

3.3 Equivalence to the Forwards Algorithm

Showing that the inverse compositional algorithm takes the same steps, to a first order approximation, as the forwards compositional algorithm is quite different to showing that the two forwards algorithms are equivalent. As mentioned in passing above, the first step is to note that the summations in Equations (7) and (20) are discrete approximations to integrals. Equation (7) is the discrete version of:

$$\int_T [\, I(\mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p})) - T(\mathbf{x})\,]^2 \; \mathrm{d}\mathbf{x} \tag{24}$$

where the integration is performed over the template T. Setting $\mathbf{y} = \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$, or equivalently $\mathbf{x} = \mathbf{W}(\mathbf{y};\Delta\mathbf{p})^{-1}$, and changing variables, Equation (24) becomes:

$$\int_{\mathbf{W}(T)} \left[\, I(\mathbf{W}(\mathbf{y};\mathbf{p})) - T(\mathbf{W}(\mathbf{y};\Delta\mathbf{p})^{-1})\,\right]^2 \left|\frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}}\right| \mathrm{d}\mathbf{y} \tag{25}$$

where the integration is now performed over the image of T under the warp $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ which we denote: $\mathbf{W}(T) = \mathbf{y} \in \{\mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \,|\, \mathbf{x} \in T\}$. Because $\mathbf{W}(\mathbf{x};\mathbf{0})$ is assumed to be the identity warp, we have:

$$\left|\frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}}\right| \; = \; 1 + O(\Delta\mathbf{p}). \tag{26}$$

The region over which integration is performed $\mathbf{W}(T) = \{\mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \,|\, \mathbf{x} \in T\}$ is equal to $T = \{\mathbf{W}(\mathbf{x};\mathbf{0}) \,|\, \mathbf{x} \in T\}$ to a zeroth order approximation also. Since we are ignoring higher order terms in $\Delta\mathbf{p}$, Equation (25) simplifies to:

$$\int_T \left[\, T(\mathbf{W}(\mathbf{y};\Delta\mathbf{p})^{-1}) - I(\mathbf{W}(\mathbf{y};\mathbf{p}))\,\right]^2 \, \mathrm{d}\mathbf{y}. \tag{27}$$

Here we assume that $T(\mathbf{W}(\mathbf{y};\Delta\mathbf{p})^{-1}) - I(\mathbf{W}(\mathbf{y};\mathbf{p}))$, or equivalently $T(\mathbf{y}) - I(\mathbf{W}(\mathbf{y};\mathbf{p}))$, is $O(\Delta\mathbf{p})$. (This assumption is equivalent to the assumption made in [Hager and Belhumeur, 1998] that the current estimate of the parameters is approximately correct.) The first order terms in the Jacobian and the area of integration can therefore be ignored. Equation (27) is then the continuous version of Equation (20) except that the term $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ is inverted. The estimate of $\Delta\mathbf{p}$ that is computed by the inverse compositional algorithm gives an estimate of $\mathbf{W}(\mathbf{x};\mathbf{p})$ that is the inverse of the warp computed by the compositional algorithm. Since the inverse compositional algorithm inverts $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ before composing it with the current estimate in Step 4, the two algorithms take the same steps to first order in $\mathbf{p}$.

Since $\mathbf{W}(\mathbf{x};\mathbf{p})$ is in general non-linear, we strictly need to point out that $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1} = \mathbf{W}(\mathbf{x};-\Delta\mathbf{p})$ to first order in $\Delta\mathbf{p}$ to fully complete the proof of equivalence. (See Section 4.1 for a derivation.) The value of $\Delta\mathbf{p}$ that is estimated by the inverse compositional algorithm is therefore the negative of what the forwards compositional algorithm estimates. This value of $\Delta\mathbf{p}$ then gives the inverse warp.

3.4 Experimental Validation

We have proved that the two forwards algorithms and the inverse compositional algorithm take the same steps to first order in $\Delta\mathbf{p}$. (The inverse additive algorithm was shown to be equivalent in [Hager and Belhumeur, 1998].) The following experiment was performed to validate the proof. We experiment with homographies to highlight the fact that the inverse compositional algorithm can be used with them. The inverse additive algorithm cannot be used on homographies, although efficient non gradient descent algorithms have been proposed [Gleicher, 1997].

We started with a 100 × 100 pixel sub-image of a larger image. (See [Baker *et al.*, 2001] for the image.) We randomly perturbed the four corners of the sub-image with 2D Gaussian translations and then solved for the homography between the perturbed corners and the originals. We next warped the original image to generate an input image for the algorithms. The three algorithms were then run with that image. As an error metric, we measured the RMS distance between the four corners of the sub-image as predicted by the computed homography and their known positions in the original image. These steps were repeated 1000 times with different random translations and the results averaged.

Figure 1 shows the convergence of the algorithms. We plot the RMS distance error in the locations of the four corners of the sub-image, averaged first over the four corners, and then over the 1000 iterations. The error is plot against the number of iterations taken by the algorithm. (The error for 0 iterations is the error in the input data.) The results in Figure 1 show that the three algorithms all converge at almost exactly the same rate validating the fact that they take approximately the same steps in each iteration. The computational cost of the inverse compositional algorithm is of course substantially less than that of the other algorithms.

4 Fitting Flexible Appearance Models

Our motivation for developing a framework for image alignment was to help develop algorithms for fitting Flexible Appearance Models[2] (FAMs) [Cootes *et al.*, 1998] and Active

[2] By Flexible Appearance Models we mean models where the shape and appearance eigenspaces are independent, that is as opposed to the closely related concept of *Active Appearance Models* (AAMs) [Cootes *et*

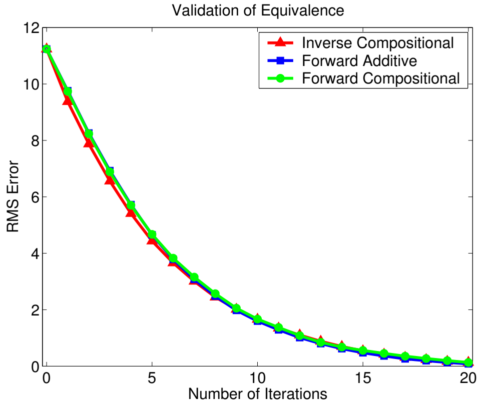


Figure 1: To validate the equivalence of the algorithms we conducted an experiment on how fast they converge. A large number of example images were generated by warping an image with randomly generated homographies. The error in the estimate of the homography is plotted against the number of iterations of the algorithm. The speed of convergence of the three algorithms is approximately the same, validating their equivalence. The computational cost of each iteration is far greater for the forwards algorithms.

Blobs [Sclaroff and Isidoro, 1998]. Both Flexible Appearance Models (FAMs) and Active Blobs are based on a combination of piecewise affine warps and appearance variation. Previously, the users of FAMs, AAMs, and Active Blobs have had to resort to "non gradient descent" algorithms to obtain efficiency. Developing a gradient descent algorithm for FAMs demonstrates the utility of our framework.

FAM fitting algorithms usually assume that there is a *constant* linear relationship between the error image and the *additive* update to the parameters. This assumption (which is equivalent to assuming that there is an efficient additive algorithm for FAMs) is incorrect. See [Matthews and Baker, 2001] for a counter-example. Difference decomposition [Gleicher, 1997] is generally used for Active Blobs, although it is also often used erroneously in the additive formulation. See Equation (19) in [Cascia *et al.*, 2000].

An FAM or Active Blob consists of four components. The first component is a template image $T(\mathbf{x})$. Typically T is an "average" image. The second component consists of a pair of triangular meshes. The first mesh is *fixed* in the coordinate frame of T. Suppose the fixed mesh has m vertices $\{(\overline{x}_i, \overline{y}_i) \mid i = 1, \ldots, m\}$. The second mesh is *flexible* and can move in the coordinate frame of the input image $I(\mathbf{x})$ and has vertices $\{(x_i, y_i) \mid i = 1, \ldots, m\}$.

When combined, the two meshes define a *piecewise affine* warp between T and I. The vertices of any pair of corresponding triangles uniquely define an affine warp between that pair of triangles. Denote the i^{th} triangle $t_i = (j, k, l)$, where $j, k, l \in \{1, \ldots, m\}$ and $t_i = (j, k, l)$ corresponds to fixed vertices $(\overline{x}_j, \overline{y}_j)$, $(\overline{x}_k, \overline{y}_k)$, and $(\overline{x}_l, \overline{y}_l)$, and flexible vertices (x_j, y_j), (x_k, y_k), and (x_l, y_l). Denote the affine warp between these two triangles $\mathbf{Affine}_{t_i}$.

The third component of the FAM is an appearance eigenspace $\{A_i(\mathbf{x}) \mid i = 1, \ldots, d\}$. As discussed in Section 2.4, the appearance eigenspace can be used to model either illumination variation or more general appearance variation. The final component of an FAM is a shape eigenspace. The shape eigenspace is defined by a set of n orthonormal shape eigen-vectors $\mathbf{s}_i$. Each shape eigenvector $\mathbf{s}_i$ is a column vector with $2 \times m$ components, one for each pair of x and y mesh vertex coordinates. The space of allowed deformations of the flexible mesh is defined by:

$$(x_1, y_1, \ldots, x_m, y_m)^{\mathrm{T}} = (\overline{x}_1, \overline{y}_1, \ldots, \overline{x}_m, \overline{y}_m)^{\mathrm{T}} + \sum_{i=1}^{n} p_i \mathbf{s}_i \tag{28}$$

The shape parameters $\mathbf{p} = (p_1, \ldots, p_n)^{\mathrm{T}}$ then define the piecewise affine warp $\mathbf{W}(\mathbf{x}; \mathbf{p})$ between the two coordinate frames. See [Matthews and Baker, 2001] for an example of an FAM and a description of how FAMs are constructed.

Unlike most warps used in computer vision, such as homographies and 3D rotations [Shum and Szeliski, 2000], the set of piecewise affine warps (onto a fixed mesh) unfortunately does not form a group and so the inverse compositional algorithm cannot be used *as is* to fit FAMs. We now extend the algorithm so that it can be used to fit FAMs. The approach is to develop first order approximations to the inverse of a warp and the composition of two warps. Since these approximations are correct to first order (the usual approximation) the extended algorithm is also correct.

4.1 Inverting the Incremental Warp

Deriving a first order approximation to $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})^{-1}$ is straightforward. A Taylor expansion gives:

$$\mathbf{W}(\mathbf{x}; \Delta\mathbf{p}) = \mathbf{W}(\mathbf{x}; \mathbf{0}) + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} = \mathbf{x} + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p}. \tag{29}$$

We therefore have:

$$\mathbf{W}(\mathbf{x}; \Delta\mathbf{p}) \circ \mathbf{W}(\mathbf{x}; -\Delta\mathbf{p}) = \mathbf{x} + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} = \mathbf{x} \tag{30}$$

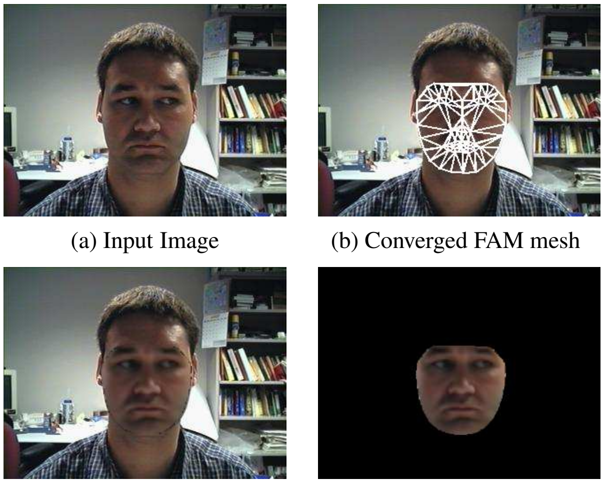
to first order in $\Delta\mathbf{p}$. Note that the two Jacobians in Equation (30) are not evaluated at exactly the same location but the results are equal to zeroth order in $\Delta\mathbf{p}$. Since the difference is multiplied by $\Delta\mathbf{p}$ we can ignore the first and higher order terms. We therefore have (to first order in $\Delta\mathbf{p}$):

$$\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})^{-1} = \mathbf{W}(\mathbf{x}; -\Delta\mathbf{p}). \tag{31}$$

4.2 Composing the Incremental Warp

We derive a first order approximation to the composition of two warps by working with the mesh vertices and approximating the destination of the fixed mesh vertices under the

al., 1998] where the allowed shape and appearance variation are coupled.



(a) Input Image (b) Converged FAM mesh

(c) Overlaid FAM Fit (d) Reconstructed Model

Figure 2: One image from a 236 frame movie (see the movie file on the CD-ROM for the complete sequence), with the results of FAM fitting using the inverse compositional algorithm.

combined warp. If the combined warp is approximately correct for the vertices (to first order in $\Delta\mathbf{p}$), it will also be approximately correct in the triangle interiors.

We wish to approximate the destination of the fixed mesh vertices $(\overline{x}_i, \overline{y}_i)$ under $\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$. Denote the destination of $(\overline{x}_i, \overline{y}_i)$ under $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$ by $(\overline{x}_i, \overline{y}_i) + (\overline{\Delta x}_i, \overline{\Delta y}_i)$. Since $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1} = \mathbf{W}(\mathbf{x};-\Delta\mathbf{p})$, Equation (28) simplifies to give:

$$\left(\overline{\Delta x}_1, \overline{\Delta y}_1, \ldots, \overline{\Delta x}_m, \overline{\Delta y}_m\right)^{\mathrm{T}} = \sum_{i=1}^{n} -\Delta p_i \mathbf{s}_i. \tag{32}$$

We next compute the change to the destination of $(\overline{x}_i, \overline{y}_i)$ under $\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$; i.e. the change from under $\mathbf{W}(\mathbf{x};\mathbf{p})$. Denote this change $(\Delta x_i, \Delta y_i)$. To compute $(\Delta x_i, \Delta y_i)$ from $(\overline{\Delta x}_i, \overline{\Delta y}_i)$, the motion of the fixed vertices $(\overline{\Delta x}_i, \overline{\Delta y}_i)$ is simply warped with an affine warp:

$$(\Delta x_i, \Delta y_i) = \mathbf{Affine_t}(\overline{\Delta x}_i, \overline{\Delta y}_i). \tag{33}$$

The triangle t to use here is the triangle that the vector $(\overline{\Delta x}_i, \overline{\Delta y}_i)$ lies in. (See [Matthews and Baker, 2001] for the details of this step which are omitted for lack of space.)

The motions of the flexible vertices $(\Delta x_i, \Delta y_i)$ are then projected into the shape eigenspace using:

$$\Delta p_i' = (\Delta x_1, \Delta y_1, \ldots, \Delta x_m, \Delta y_m)\, \mathbf{s}_i \tag{34}$$

where $\Delta\mathbf{p}'$ is the modified vector of parameter increments that when added to $\mathbf{p}$ gives $\mathbf{p} + \Delta\mathbf{p}'$ as the parameters of $\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$. In summary, Equations (32), (33), and (34) can be used to compute the parameters $\mathbf{p} + \Delta\mathbf{p}'$ of $\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$ from $\mathbf{p}$ and $\Delta\mathbf{p}$. The computational cost of this step is $O(nm)$ which is negligible compared to Steps (1–3) of the inverse compositional algorithm.

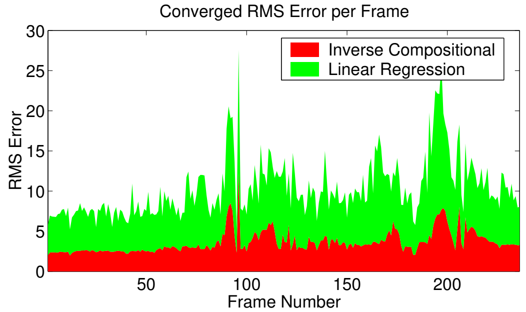


Figure 3: The error in the final FAM fit plotted across the entire 236 frame sequence. The inverse compositional algorithm is able to fit the FAM far better, resulting in a much lower RMS error.

4.3 Experimental Results

There are two differences between our FAM fitting algorithm and previous ones: (1) our algorithm is an analytically derived gradient descent algorithm, rather than using numerical techniques such as *linear regression* [Cootes *et al.*, 1998], *finite differences* [Cootes and Taylor, 2001], or *difference decomposition* [Gleicher, 1997, Cascia *et al.*, 2000], and (2) we update the new estimate of the warp using the inverse compositional algorithm rather than simply adding the parameter increments. As we showed in [Matthews and Baker, 2001] (and was mentioned in passing in [Gleicher, 1997]) the naive additive approach is provably wrong.

4.3.1 Comparison with other FAM Fitting Algorithms

We first compare our algorithm with the original regression-based AAM algorithm [Cootes *et al.*, 1998] (applied to FAMs), on a sequence of 236 frames. One example input frame, the FAM, the converged FAM overlaid on the input, and the result of fitting are shown in Figure 2. (A movie of the FAM being fit over the entire sequence is contained on the CD-ROM version of the proceedings.)

Figure 3 plots the RMS pixel error between the final FAM fit and the input image, for each of the 236 frames in the sequence. Although the models used are exactly the same, the inverse compositional algorithm is able to fit far better. The error in the fit (which is partly due to the fact that the model may not completely explain the data anyway) is far lower for the inverse compositional algorithm than for the regression-based algorithm of [Cootes *et al.*, 1998]. The effect of this improved fitting on the movie on the CD-ROM is that the model fit looks far smoother across time.

4.3.2 Comparison with the Naive Additive Algorithm

To demonstrate the importance of the compositional framework, we compared the inverse compositional algorithm with another gradient descent algorithm that is identical except that it naively updates the warp by adding the parameter increments rather than using the inverse compositional algorithm. The evaluation is on a task in the automatic construction of FAMs outlined in [Matthews and Baker, 2001].

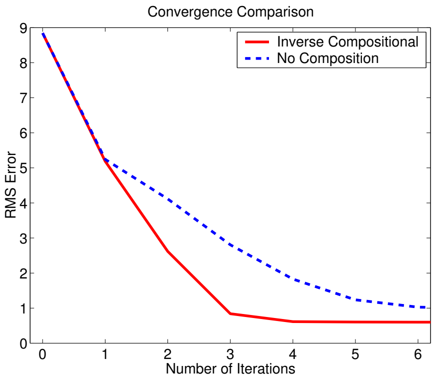


Figure 4: A demonstration of the importance of using the inverse compositional algorithm on a task in the automatic construction of FAMs. Naively adding the parameter updates (assuming a constant Hessian) rather the composing the incremental warp results in much slower convergence and a worse final fit.

Figure 4 contains a plot of the error in the FAM fit against the number of iterations. The figure demonstrates that with the inverse compositional algorithm, both the convergence rate is faster, and the final converged fit is better. Without using the compositional framework, the algorithm does converge, albeit slowly, because naively updating the parameters additively corresponds to taking gradient descent steps in approximately the right direction, but not quite the optimal direction; i.e. it converges "by chance". The convergence rate is almost twice as fast. The naive algorithm takes over 6 iterations to reach the same degree of fit that the inverse compositional algorithm reaches in 3.

5 Discussion

We have presented a framework (see Table 1) for gradient descent image alignment. Algorithms can either be *additive* or *compositional*, and either *forwards* or *inverse*. The forwards additive algorithm [Lucas and Kanade, 1981], the inverse additive algorithm [Hager and Belhumeur, 1998], and the forwards compositional algorithm [Shum and Szeliski, 2000] have all been studied before. The inverse compositional algorithm and its extension to piecewise affine warps follow directly from the framework.

Due to lack of space, we are unable to present the full details of our experiments in this paper. More details can be found in the associated technical report [Matthews and Baker, 2001]. We are also currently conducting an extensive evaluation of FAM and AAM fitting algorithms.

Acknowledgments

We would like to thank Bob Collins, Frank Dellaert, Takeo Kanade, Jianbo Shi, Sundar Vedula, and Jing Xiao for discussions on image alignment. We would also like to thank the anonymous reviewers for their feedback. The research described in this paper was conducted under U.S. Office of Naval Research contract N00014-00-1-0915.

Table 1: Gradient descent image alignment algorithms can either be *additive* or *compositional*, and either *forwards* or *inverse*. Our framework leads immediately to two new algorithms, the *inverse compositional* algorithm and its extension for fitting FAMs.

Algorithm	Can be Applied To	Efficient?
Forwards Additive	Any	No
Inverse Additive	Simple Linear 2D +	Yes
Forwards Compositional	Any Semi-Group	No
Inverse Compositional	Any Group	Yes
FAM Fitting Algorithm	Piecewise Affine	Yes

References

[Baker *et al.*, 2001] S. Baker, F. Dellaert, and I. Matthews. Aligning images incrementally backwards. Technical Report CMU-RI-TR-01-03, CMU Rob. Institute, 2001.

[Bergen *et al.*, 1992] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. of ECCV*, pages 237–252, 1992.

[Black and Jepson, 1998] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 36(2):101–130, 1998.

[Cascia *et al.*, 2000] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE PAMI*, 22, 2000.

[Cootes and Taylor, 2001] T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision. World Wide Web Publication, February 2001. (Available from http://www.isbe.man.ac.uk/˜bim/refs.html).

[Cootes *et al.*, 1998] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *Proc. of ECCV*, volume 2, pages 484–498, 1998.

[Dellaert and Collins, 1999] F. Dellaert and R. Collins. Fast image-based tracking by selective pixel integration. In *Proc. of the ICCV Wkshp on Frame-Rate Vision*, 1999.

[Gleicher, 1997] M. Gleicher. Projective registration with difference decomposition. In *Proc. of CVPR*, 1997.

[Hager and Belhumeur, 1998] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE PAMI*, 20, 1998.

[Lucas and Kanade, 1981] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of IJCAI*, pages 674–679, 1981.

[Matthews and Baker, 2001] I. Matthews and S. Baker. Fitting flexible appearance models. Technical Report CMU-RI-TR-01-04, CMU Robotics Institute, 2001.

[Sclaroff and Isidoro, 1998] S. Sclaroff and J. Isidoro. Active blobs. In *Proc. of ICCV*, 1998.

[Shum and Szeliski, 2000] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *IJCV*, 16(1):63–84, 2000.