Aligning Images Incrementally Backwards

Simon Baker, Frank Dellaert, and Iain Matthews

CMU-RI-TR-01-03

Abstract

We propose an efficient algorithm for image alignment. The key step is to estimate the *inverse* of the incremental update to the warp. The incremental update is then *inverted* and *composed* with the current estimate to form the new estimate for the next iteration. The set of warps must therefore form a *group.* The algorithm is efficient because estimating the inverse of the update reverses the role of the two images. Much of the computation then only needs to be performed once, rather than once for each iteration.

Keywords: Backwards incremental image alignment and registration, efficiency.

1 Introduction

Image *registration* or *alignment* 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 (or layered) motion estimation [Bergen *et al.*, 1992], mosaic-ing [Shum and Szeliski, 2000], and face coding [Lanitis *et al.*, 1997]. The template is often taken to be (a sub-region of) another image and the underlying goal is to compute correspondences between the two images. The template can also include a model of appearance variation, whether it be a model of illumination variation [Cascia *et al.*, 2000], or a more general model of appearance variation [Black and Jepson, 1998].

The efficiency of image alignment has of course been studied before. Most notably [Hager and Belhumeur, 1998] proposed an efficient algorithm for translations, 2D similarity transforms, and affine warps. The derivation of their algorithm, however, is fairly complex and requires that the warp have a particularly simple form. As a result, their algorithm unfortunately cannot be used with many common warps such as homographies and 3D rotations [Shum and Szeliski, 2000] and piecewise affine warps [Lanitis *et al.*, 1997].

We propose a general-purpose algorithm for efficient image alignment. Following the approach in [Shum and Szeliski, 2000], we solve for an incremental update to the current estimate of the warp. This makes the Jacobian a constant. On its own, however, this is not enough to make the algorithm much more efficient. The key step that we propose is to estimate the *inverse* of what we want; i.e. the inverse of the incremental update to the warp. This step reverses the roles of the two images. Since the template image is fixed in its own coordinate frame, the image gradients are then also constant. An algorithm that is as efficient as that proposed in [Hager and Belhumeur, 1998] results. (Our algorithm, like theirs, can be extended to allow linear appearance variation at little extra online cost.)

Besides being simpler and therefore easier to implement, our algorithm can also be

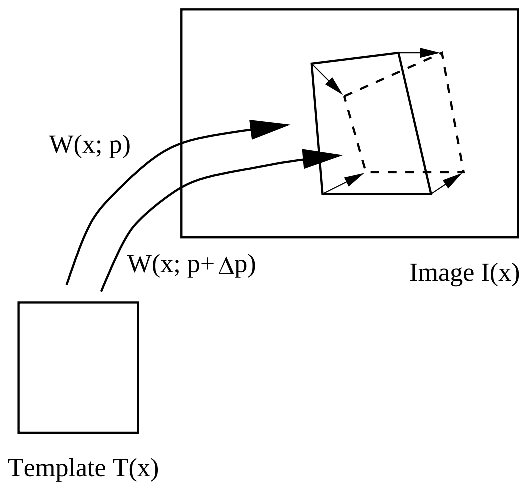


Figure 1: The traditional approach to image alignment consists of iteratively solving for incremental updates $\Delta\mathbf{p}$ to the current estimate of the warp parameters $\mathbf{p}$. This algorithm can be very inefficient because both the image gradients and the Jacobian change from iteration to iteration.

used with a much wider class of warps. In particular, it can be used with homographies and 3D rotations. Moreover, we can precisely describe the warps that the algorithm can be used with. The one and only requirement is that the set of warps form a *group*. This property is needed because the incremental update to the warp must be *inverted* and then *composed* with the current estimate to obtain the new estimate of the warp. (To use the incremental approach [Shum and Szeliski, 2000], the warps need to form a *semi-group*.)

2 Algorithms

Suppose we are trying to align a template image $T(\mathbf{x})$ to a fixed 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, 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 over the pixels in the template image $T(\mathbf{x})$. The usual approach (Figure 1) is to suppose that a current estimate of $\mathbf{p}$ is known and then to iteratively solve for increments to the parameters $\Delta\mathbf{p}$; i.e. minimize:

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

with respect to $\Delta\mathbf{p}$. This is a least squares problem, the solution of which is:

$$\Delta\mathbf{p} \;=\; \sum_{\mathbf{x}} H^{-1} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))] \tag{3}$$

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

$$H \;=\; \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]. \tag{4}$$

In the above, the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x};\mathbf{p})$ and the image gradient ∇I at $\mathbf{W}(\mathbf{x};\mathbf{p})$. Both of these quantities therefore depend upon $\mathbf{p}$ (in general.) The traditional image alignment algorithm therefore consists of iterating the following steps until the estimates of the parameters $\mathbf{p}$ converge (or equivalently $\Delta\mathbf{p}$ becomes approximately $\mathbf{0}$):

1. Warp I backwards onto the coordinate frame of T to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$;

2. Subtract $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ from $T(\mathbf{x})$ to compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$;

3. Warp the gradient of image I onto the coordinate frame of T to compute ∇I;

4. Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at the current estimate of the parameters $\mathbf{p}$;

5. Compute the Hessian matrix using Equation (4);

6. Compute $\Delta\mathbf{p}$ using Equation (3);

7. Update the current estimate of the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$.

Steps (3), (4), and (5) must, in general, be repeated for each iteration because the current estimate of the parameters $\mathbf{p}$ varies from iteration to iteration. The (backwards warped) gradient ∇I and the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ are therefore different in each iteration.

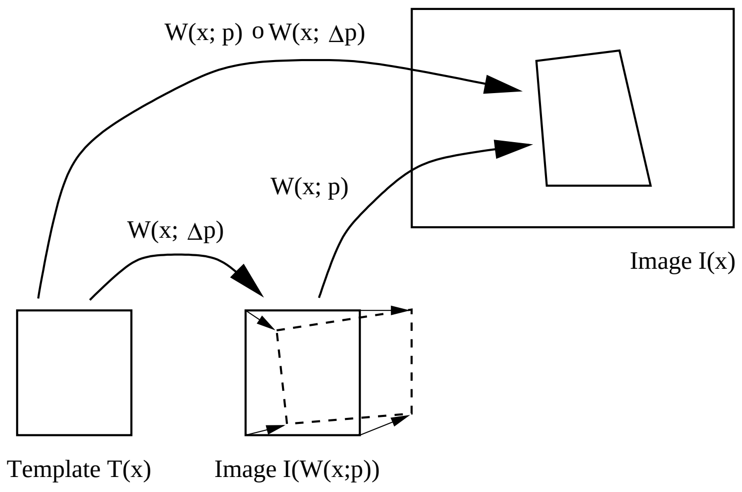


Figure 2: A modification of image alignment consists of iteratively estimating an incremental warp $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$ rather than a direct incremental update to the parameters. The advantage of this approach is that the Jacobian is usually simpler (and a constant) [Shum and Szeliski, 2000].

2.1 Incremental Image Alignment

Figure 2 shows a refinement of the previous algorithm where an incremental update warp $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$ is estimated rather than a direct incremental update to the parameters; i.e. we minimize:

$$\sum_{\mathbf{x}} [\, I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \Delta\mathbf{p}); \mathbf{p})) - T(\mathbf{x})\,]^2 \;\approx\; \sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \mathbf{0}); \mathbf{p})) + \boldsymbol{\nabla} I(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 \tag{5}$$

with respect to $\Delta\mathbf{p}$. Assuming that $\mathbf{W}(\mathbf{x}; \mathbf{0}) = \mathbf{x}$; i.e. $\mathbf{W}(\mathbf{x}; \mathbf{0})$ is the identity warp, then $I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \mathbf{0}), \mathbf{p})) = I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$. There are then two major differences between Equations (2) and (5). 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 notation. In Equation (2) the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x}; \mathbf{p})$, but in Equation (5) it is evaluated at $(\mathbf{x}; \mathbf{0})$ because that is where the first order 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. As can be seen, the

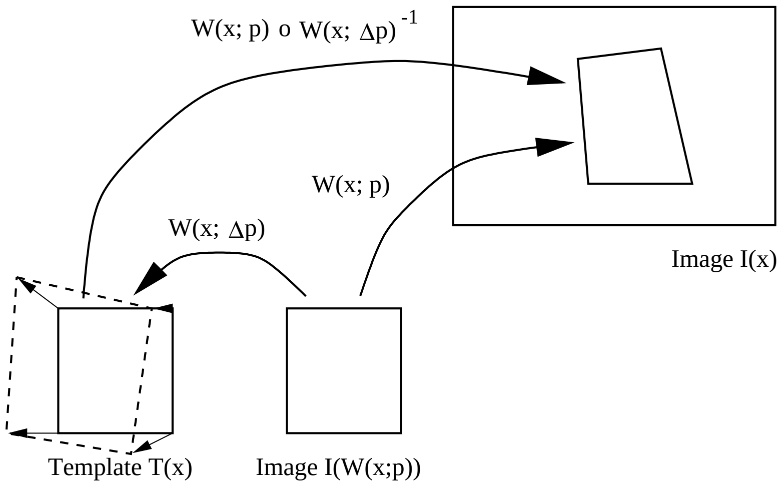


Figure 3: A refinement to the incremental algorithm consists of estimating the inverse of the incremental warp. This reverses the roles of the two images. Step (3) of the algorithm is therefore not needed. Moreover, Steps (4–5) need only be performed once, rather than once per iteration.

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 might 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 $\mathbf{W}(\mathbf{x};\mathbf{0}) = \mathbf{x}$.)

2.2 Backwards Incremental Image Alignment

A further refinement is to estimate the inverse of the incremental warp (Figure 3). In the incremental formulation, the images that are being aligned are $T(\mathbf{x})$ and $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$. Because these two images share the coordinate frame of T, their roles are symmetric and can be switched. (See Section 2.3 and Appendix A for more detailed arguments of why the roles of $T(\mathbf{x})$ and $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ can be reversed.) We therefore minimize:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{W}(\mathbf{x};\Delta\mathbf{p})) \right]^2 \approx \sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{W}(\mathbf{x};\mathbf{0})) - \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \right]^2 \quad (6)$$

with respect to $\Delta\mathbf{p}$. Assuming again that $\mathbf{W}(\mathbf{x};\mathbf{0})$ is the identity warp, the solution is:

$$\Delta\mathbf{p} \;=\; \sum_{\mathbf{x}} H^{-1}\left[\boldsymbol{\nabla} T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^{\mathrm{T}}\left[I(\mathbf{W}(\mathbf{x};\mathbf{p}))-T(\mathbf{x})\right] \tag{7}$$

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

$$H \;=\; \sum_{\mathbf{x}}\left[\boldsymbol{\nabla} T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^{\mathrm{T}}\left[\boldsymbol{\nabla} T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]. \tag{8}$$

and the Jacobian $\frac{\partial\mathbf{W}}{\partial\mathbf{p}}$ is evaluated at $(\mathbf{x};\mathbf{0})$. Since there is nothing in $\left[H^{-1}\boldsymbol{\nabla} T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]$ that depends upon $\mathbf{p}$, it is constant across iterations and can be pre-computed. Once this is done, the algorithm then becomes iterating the following four steps until convergence:

1. Warp I backwards onto the coordinate frame of T 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 (7).
4. Update the current estimate of the warp $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p})\circ\mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$.

This algorithm is much more efficient than the original algorithm. Steps (3-5) of the original algorithm 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.) This algorithm is almost exactly as efficient as the one proposed in [Hager and Belhumeur, 1998]. It can, however, be applied to any warps that form a *group* (including homographies and 3D rotations), instead of only to a small collection of warps. Our algorithm is also conceptually far simpler.

2.3 Comparison with [Hager and Belhumeur, 1998]

A question which arises at this point is why not just reverse the roles of the image and the template in the original formulation (Figure 1)? Can we not perform a change of variables

$\mathbf{y} = \mathbf{W}(\mathbf{x};\mathbf{p})$ or $\mathbf{x} = \mathbf{W}(\mathbf{y};\mathbf{p})^{-1}$? The reason we cannot do this is that, because the summation in Equation (1) is a discrete approximation to an integral, we have to incorporate the Jacobian (with respect to $\mathbf{y}$) of the warp $\mathbf{W}(\mathbf{y};\mathbf{p})^{-1}$. Equation (1) therefore becomes:

$$\sum_{\mathbf{y}} \left| \frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}} \right| \cdot \left[I(\mathbf{y}) - T(\mathbf{W}(\mathbf{y};\mathbf{p})^{-1}) \right]^2 \tag{9}$$

where the summation is over the subregion of I that corresponds to the template T warped with $\mathbf{W}(\mathbf{x};\mathbf{p})$, rather than simply:

$$\sum_{\mathbf{y}} \left[I(\mathbf{y}) - T(\mathbf{W}(\mathbf{y};\mathbf{p})^{-1}) \right]^2. \tag{10}$$

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 in the summation.)

Fortunately we do not have a similar problem in the incremental case because both Jacobians $\frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}}$ and $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ are evaluated at $\mathbf{p} = \mathbf{0}$; i.e. at the identity warp. The determinant of the Jacobian (with respect to $\mathbf{y}$) of the identity warp is 1, or more generally:

$$\left| \frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}} \right| \;=\; 1 + O(\Delta \mathbf{p}). \tag{11}$$

in a small neighborhood around $\mathbf{W}(\mathbf{x};\mathbf{0})$. Since the error image $[\, I(\mathbf{y}) - T(\mathbf{W}(\mathbf{y};\mathbf{p})^{-1})\,]$ is assumed to be $O(\Delta\mathbf{p})$ and we are ignoring higher order powers of $\Delta\mathbf{p}$, we can assume that this Jacobian is exactly 1. (The slight change in the region of summation can be ignored for the same reason.) Equations (5) and (6) are therefore *equivalent* to first order in $\Delta\mathbf{p}$. (See Appendix A for a more complete argument of why the roles of I and T can be switched.)

Note that it is sometimes assumed that the minimum values of Equations (9) and (10) are attained at approximately the same value of $\mathbf{y}$ and Equation (10) is used without further comment. While this may often be a reasonable assumption, Equation (11) shows that it is always a valid assumption for the incremental warps used in our algorithm.

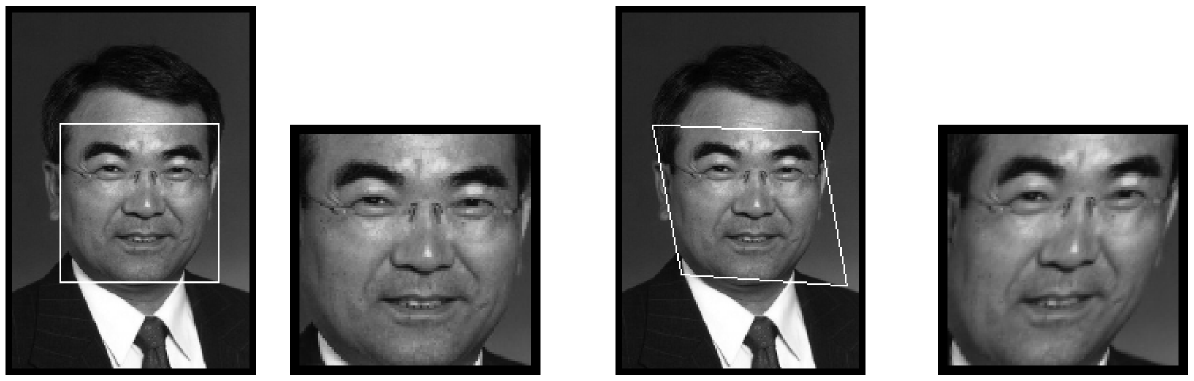
2.4 Efficiency

Comparing the efficiency of algorithms is always difficult. Even giving the run-time in seconds can be misleading. There are often a large number of "approximations" that can be made to improve the efficiency of computer vision algorithms. For example, in Section 4.1 of [Shum and Szeliski, 2000] the authors make the approximation that the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is locally constant to speed up their algorithm. In [Dellaert and Collins, 1999] an algorithm is proposed that only uses a (carefully chosen) subset of the pixels in $T(\mathbf{x})$ to achieve real-time tracking. The only points we wish to make about efficiency are the following:

- The backwards incremental algorithm is about as simple as is possible. The error image is computed, dot-producted with constant (pre-computable) images to give $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$, and then the incremental warp inverted and composed with the current estimate.
- Both of the two main steps can often be accelerated in hardware, the error image computation using a graphics card, the dot-products using MMX-like instructions.
- The algorithm is computationally almost exactly the same as [Hager and Belhumeur, 1998]. The advantage is that it can be applied to a much larger class of warps.
- Modelling appearance variation (see Appendix B for a description of how the incremental backwards algorithm can be extended to deal with appearance variation) makes efficiency much harder to achieve. In the past, various (arguably incorrect) simplifying assumptions have been made [Lanitis *et al.*, 1997, Cascia *et al.*, 2000] for sets of warps not covered by the algorithm proposed in [Hager and Belhumeur, 1998].

3 Experiments

We have described three formulations of image alignment: (1) the traditional non-incremental formulation in Equation (2), (2) the traditional incremental formulation in Equation (5), and

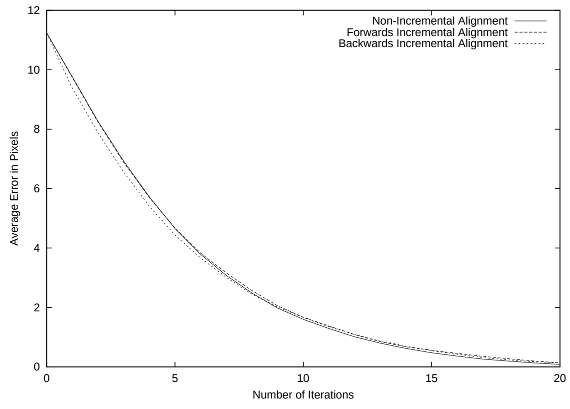


(a) Original Image (b) Cropped Region (c) Random Quadrilateral (d) Input Image

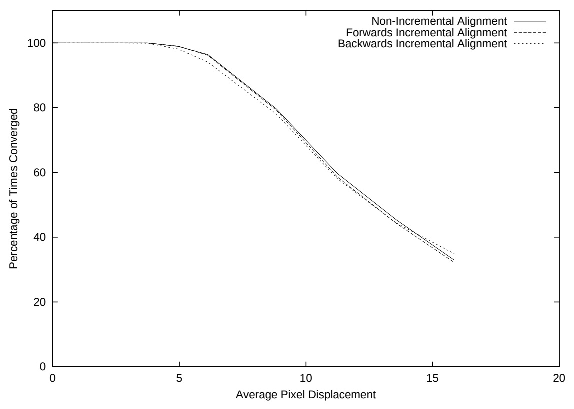
Figure 4: We conducted a synthetic experiment to validate our algorithm. We took an image (a), shown with a four corner box super-imposed on it. We then randomly translated the corners of the box using a 2D Gaussian distribution (c). We used the homography between the boxes in (a) and (c) to warp the original image (a) and generate a new image (d). The image in (d) was then passed an as input to the 3 algorithms and the homography estimated. The speed of convergence of the algorithms was then estimated by averaging over 1000 random samples like the one in (d).

(3) the backwards incremental formulation in Equation (6). Since we can show the three formulations to be equivalent to first order in $\Delta\mathbf{p}$ (see Appendix A), the three formulations are all simply re-parameterizations of each other. Re-parameterizing a non-linear optimization can affect the speed of convergence. The effect of any particular re-parameterization is fairly difficult to predict however. For example, the homography can be parameterized using the usual eight parameters [Shum and Szeliski, 2000] or instead by the location of four image points. Which is better? Although intuitively the parameterization in terms of four image points appears to be the more natural, it has not been demonstrated that this is indeed the case. Moreover, the eight parameter approach is used almost exclusively in the literature.

We performed the following synthetic experiment to show that the convergence rates of the three formulations are roughly the same. We started with the image in Figure 4(a) and the four corner points of the (100 pixel $\times$ 100 pixel) box super-imposed on that image. We randomly perturbed the four corners of the box with 2D Gaussian translations (to give Figure 4(c)) and then solved for the homography between the perturbed points and the originals. We next warped the original image to generate an input image for the algorithms



(a) Speed of Convergence



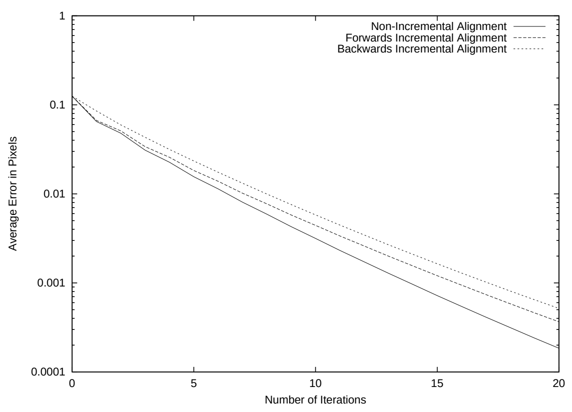
(b) Robustness of Convergence

Figure 5: The speed of convergence (a) of the three algorithms is approximately the same. The robustness of the algorithms (b) is also similar. The percentage of the time that the algorithms converge drops off rapidly after the average perturbation to the box corners increases over 5 pixels.

(see Figure 4(d).) The three algorithms are then run with that input image. As an error metric, we measured the mean distance between the four corners of the box 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.

In Figure 5(a) we plot the speed of convergence of the three algorithms. The estimated homography (which was parameterized using the standard eight parameters [Shum and Szeliski, 2000] for consistency with the literature) can be used to infer the corners of the box which are then subtracted from the known positions in the original image. We plot the distance error in the positions, averaged first over the four corners, and then over the 1000 iterations. The error is plotted 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 5(a) show that the three algorithms all converge at almost exactly the same rate.

In Figure 5(b) we plot a curve of how frequently the algorithms converged and how frequently they diverged. We say that the algorithm has converged if the error in the position of every corner of the box is less than 1 pixel after 25 iterations of the algorithm. In Figure 5(b) we plot the percentage of times the algorithms converged against the average



(a) Speed of Convergence Close to the Correct Answer

Figure 6: For small perturbations, the difference in the speeds of convergence of the algorithms becomes more noticeable. The speed of convergence close to the correct answer is also highly dependent on sampling issues such as the size of the gradient kernel. The practical importance of these results is minimal. Getting such sub-pixel accuracy is only possible with very low noise.

perturbation in the four corners of the box. We see that the behavior of all the algorithms is approximately the same. The frequency of convergence drops off rapidly after the average perturbation increases past about 5 pixels. At first glance this figure may appear somewhat low. To allow us to compare the raw performance of the algorithms, however, we chose not to include any of the usual refinements that can be made to increase both the robustness and the speed of convergence of alignment algorithms. These refinements, which include: (1) running the algorithm hierarchically on a image pyramid, (2) using increasing transformation complexity, and (3) iteratively resizing the step size based on whether the algorithm moved towards the solution or not (c.f. Levenberg-Marquardt), can all be used with any of the algorithms described in this paper. There is no reason to believe they would help any formulation of the problem more than any other and so for simplicity we use none.

Very close to the correct answer, the situation is a little different. In Figure 6 we plot results of running the algorithms with very small (≈ 0.1 pixels) perturbations. (The average error on the y-axis is plotted on a log scale so that the difference can be seen more clearly.) Close to the correct answer the algorithms perform noticeably differently, with the traditional non-incremental algorithm performing the best. The exact cause of this difference is hard to identify. The influence of the choice of the size of the gradient kernel and the details of the interpolation algorithm are very large close to the correct answer. Part of the difference is probably caused by the re-parameterization, part due to other sampling effects. The results in Figure 6 are of little practical importance. The image noise will generally make getting sub-pixel accuracies than are significantly less than 0.1 pixels impossible anyway.

4 Conclusion

The main contribution of this paper is to point out that the role of the two images can be switched in the incremental formulation of image alignment. (See Appendix A for more details.) The determinant of the Jacobian (with respect to the image coordinates) which must be introduced when changing variables is 1 (to a first order approximation.) It can therefore be ignored, unlike in the non-incremental formulation [Hager and Belhumeur, 1998]. This fact was used to develop an efficient algorithm for image alignment that can be used with any set of warps which form a group. Unfortunately there are sets of warps which do not form a group, perhaps most notably the set of piecewise affine warps (onto a fixed mesh) [Lanitis *et al.*, 1997]. We are currently working on extending our algorithm so that it can be used with such piecewise affine warps. The approach is to develop first order (in $\Delta\mathbf{p}$) approximations to the inverse warp $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$ and the composition $\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$.

Acknowledgements

We would like to thank Bob Collins, Takeo Kanade, Jianbo Shi, Sundar Vedula, and Jing Xiao for discussions on image alignment. The research described in this report was conducted under U.S. Office of Naval Research (ONR) contract N00014-00-1-0915.

References

[Bergen *et al.*, 1992] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the Second European Conference on Computer Vision*, pages 237–252, Santa Margherita Liguere, Italy, 1992.

[Black and Jepson, 1998] M.J. 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 Transactions on Pattern Analysis and Machine Intelligence*, 22(4), 2000.

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

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

[Lanitis *et al.*, 1997] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):742–756, 1997.

[Lucas and Kanade, 1981] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, British Columbia, 1981.

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

A First Order Equivalence of the Three Formulations

We have described three different formulations of image alignment: (1) the traditional *non-incremental* formulation, which is to minimize:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{x}) \right]^2, \tag{12}$$

(2) the traditional *forwards incremental* formulation, which is to minimize:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p})) - T(\mathbf{x}) \right]^2, \tag{13}$$

and (3) the *backwards incremental* formulation, which is to minimize:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{W}(\mathbf{x};\Delta\mathbf{p})) \right]^2. \tag{14}$$

The minimization is performed with respect to $\Delta\mathbf{p}$ in all three case. The update to the current estimate of the warp is different, however. In the traditional non-incremental formulation, the update is:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \;\leftarrow\; \mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}), \tag{15}$$

in the traditional forwards incremental formulation, the update is:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}), \tag{16}$$

and in the backwards incremental formulation, the update is:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}. \tag{17}$$

We now show that these three formulations are equivalent in the sense that the updated warps in Equations (15–17) are equal to first order in $\Delta\mathbf{p}$. We start by showing that the non-incremental and forwards incremental formulations are equivalent. Afterwards, we show that the two incremental formulations are equivalent. The fact that the non-incremental and the backwards incremental formulations are equivalent then follows by transitivity.

A.1 Non-Incremental and Forwards Incremental Formulations

Equation (12) simplifies (as shown in Section 2) to:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 \tag{18}$$

when a first order Taylor expansion is made in $\Delta\mathbf{p}$. Similarly, the update to the warp in Equation (15) simplifies to:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \;\leftarrow\; \mathbf{W}(\mathbf{x};\mathbf{p}) + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \tag{19}$$

where, again, a first order Taylor expansion is made for $\Delta\mathbf{p}$. In the forwards incremental formulation, Equation (13) simplifies to:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x};\mathbf{0});\mathbf{p})) + \nabla I(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 . \tag{20}$$

where $\nabla I(\mathbf{W})$ is the gradient of $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$, which equals $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{x}}$ by the chain rule. Assuming that $\mathbf{W}(\mathbf{x};\mathbf{0})$ is the identity warp, Equation (20) simplifies further to:

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

To simplify the update for the forwards incremental formulation in Equation (16) note that:

$$\mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \;\approx\; \mathbf{W}(\mathbf{x};\mathbf{0}) + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \;=\; \mathbf{x} + \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \tag{22}$$

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

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \;=\; \mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p}). \tag{23}$$

Combining these last two equations (and applying the Taylor expansion again) gives the update for the forwards incremental formulation as:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \;\leftarrow\; \mathbf{W}(\mathbf{x};\mathbf{p}) + \frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p}. \tag{24}$$

As can be seen, the only difference between the non-incremental formulation in Equations (18) and (19), and the forwards incremental formulation in Equations (21) and (24), is that the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is replaced by $\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$. (In the second expression $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x}; \mathbf{0})$, rather than $(\mathbf{x}; \mathbf{p})$, so the $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ terms are, in general, different.) Because of this difference, Equations (18) and (21) generally result in different estimates for $\Delta \mathbf{p}$ in the two formulations. The overall update to the warp, however, $\frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p}$ in the non-incremental formulation and $\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p}$ in the forwards incremental formulation, are the same to first order in $\Delta \mathbf{p}$. The reason is as follows. The vectors $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ in the non-incremental formulation and $\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ in the forwards incremental formulation both span the same linear space; in particular, they both span the tangent space to the set of warps $\mathbf{W}(\mathbf{x}; \mathbf{p})$. Since these vectors span the same space, the optimal value of $\frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p}$ in Equation (18) must equal the optimal value of $\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p}$ in Equation (21). The updates are therefore the same (to first order in $\Delta \mathbf{p}$.)

A.2 Forwards and Backwards Incremental Formulations

The derivation of the equivalence of the two incremental formulations is very different to that of the non-incremental formulation and the forwards incremental formulation. As argued in Section 2.3, the first step is to note that the summations in Equations (13) and (14) can be thought of as discrete approximations to integrals. Equation (13) is the discrete version of:

$$\int_{\mathbf{x} \in T} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p}); \mathbf{p})) - T(\mathbf{x}) \right]^2 \, d\mathbf{x} \tag{25}$$

where the integration is performed over the template image T. If we set $\mathbf{y} = \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})$ or equivalently $\mathbf{x} = \mathbf{W}(\mathbf{y}; \Delta \mathbf{p})^{-1}$ and perform a change of variables, Equation (25) becomes:

$$\int_{\mathbf{y} \in \{\mathbf{W}(\mathbf{x}; \Delta \mathbf{p}) \,|\, \mathbf{x} \in T\}} \left[I(\mathbf{W}(\mathbf{y}; \mathbf{p})) - T(\mathbf{W}(\mathbf{y}; \Delta \mathbf{p})^{-1}) \right]^2 \left| \frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}} \right| d\mathbf{y} \tag{26}$$

where the integration is now performed over the image of T under the warp $\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})$. As argued in Section 2.3:

$$\left| \frac{\partial \mathbf{W}^{-1}}{\partial \mathbf{y}} \right| \; = \; 1 + O(\Delta \mathbf{p}) \tag{27}$$

because $\mathbf{W}(\mathbf{x};\mathbf{0})$ is assumed to be the identity warp. Also, the region over which integration is performed $\{\mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \mid \mathbf{x} \in T\}$ is equal to $T = \{\mathbf{W}(\mathbf{x};\mathbf{0}) \mid \mathbf{x} \in T\}$ to a zeroth order approximation. Since we are ignoring higher order terms in $\Delta\mathbf{p}$, Equation (26) simplifies to:

$$\int_{\mathbf{y}\in T} \left[I(\mathbf{W}(\mathbf{y};\mathbf{p})) - T(\mathbf{W}(\mathbf{y};\Delta\mathbf{p})^{-1}) \right]^2 \, \mathrm{d}\mathbf{y}. \tag{28}$$

In making this simplification, we have assumed that $I(\mathbf{W}(\mathbf{y};\mathbf{p})) - T(\mathbf{W}(\mathbf{y};\Delta\mathbf{p})^{-1})$ (or equivalently $I(\mathbf{W}(\mathbf{y};\mathbf{p})) - T(\mathbf{y})$) is $O(\Delta\mathbf{p})$. The first order terms in the Jacobian and the area of integraton can therefore be ignored. Equation (28) is then the continuous version of Equation (14) modulo the the inversion of $\mathbf{W}$. The inversion of $\mathbf{W}$ can be ignored because:

$$\mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1} \;=\; \mathbf{W}(\mathbf{x};-\Delta\mathbf{p}) \tag{29}$$

to first order in $\Delta\mathbf{p}$.

B Extension: Modelling Appearance Variation

Often it is assumed that the template $T(\mathbf{x})$ 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 [Lanitis *et al.*, 1997, Black and Jepson, 1998]. Suppose that the images $A_1(\mathbf{x}), \ldots, A_d(\mathbf{x})$ are an (orthonormal) basis for the linear subspace. The alignment problem is then posed as one of minimizing:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) + \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right]^2 \;=\; \left\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) + \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right\|^2 \tag{30}$$

where the minimization is now conducted simultaneously over both the vector of parameters $\mathbf{p}$ and the 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 (30) can be rewritten as:

$$\left\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) + \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right\|^2_{\mathrm{span}(A_i)^{\perp}} + \left\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) + \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right\|^2_{\mathrm{span}(A_i)} \tag{31}$$

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

$$\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) \|^2_{\text{span}(A_i)^\perp} + \left\| I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) + \sum_{i=1}^{d} \lambda_i A_i(\mathbf{x}) \right\|^2_{\text{span}(A_i)} . \tag{32}$$

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. Therefore the minimum value can be found sequentially by first minimizing the first term with respect to $\mathbf{p}$ alone, and then using that optimal value of $\mathbf{p}$ as a constant to minimize the second term with respect to the λ_i. Assuming that the basis vectors A_i are orthonormal, the second minimization has a simple closed-form solution:

$$\lambda_i \;=\; -\sum_{\mathbf{x}} A_i(\mathbf{x}) \cdot \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) \right], \tag{33}$$

the dot product of A_i with the final error image obtained after doing the first minimization.

Minimizing the first term in Equation (32) is not really any different to solving the original alignment problem. We just need to work in the linear subspace $\text{span}(A_i)^\perp$. We do not even need to project the error image into this subspace. All we need to do is project $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ into that subspace in Equations (3) and (4). (The error image does not need to be projected into this subspace because Equation (3) is really the dot product of the error image with $H^{-1} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}}$. So long as one of the two terms in a dot product is projected into a linear subspace, the result is the same as them both being projected into it.)

All of this appearance variation analysis applies equally to the "backwards incremental" algorithm. The only modification that is needed is to project $\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ into $\text{span}(A_i)^\perp$ in Equations (7) and (8). (Similarly to Section 2.3 and Appendix A, the subspace $\text{span}(A_i)^\perp$ moves slightly in the coordinate frame of $T(\mathbf{x})$ during the change of coordinates, however the correction is $O(\Delta\mathbf{p})$ and can be ignored for the same reason we ignored the higher order terms in the determinant of the Jacobian in Equation (11).)

In summary, the algorithm with linear appearance variation is exactly the same as without except: (1) $\boldsymbol{\nabla} T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ must be projected into the subspace orthogonal to $\mathrm{span}(A_i)$ before being used in Equations (7) and (8), and (2) once the parameters $\mathbf{p}$ have been estimated, Equation (33) is used to estimate the appearance parameters λ_i if so desired. (Often the values of the λ_i are not required.) The projection into the subspace orthogonal to $\mathrm{span}(A_i)$ could be written as a single (large) matrix product, but in practice it is computed by projecting out the component in each of the orthonormal directions $A_1, \ldots, A_d$ in turn.