Image Alignment With Appearance Variation

Image *registration* or *alignment* consists of moving, and possibly deforming, a template to minimize the difference between the template and an image. (See Figure 1.) Some of the numerous applications of registration include optical flow [7], tracking [3, 4], parametric (layered) motion estimation [1], super-resolution [5], mosaic-ing [8], and face coding using active appearance models (AAMs) [6]. In many cases, the template is (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 simple model of illumination variation [3], or a more general model of variation [6]. The transformations (i.e. motion and deformations) of the template that are allowed vary from application to application, and from simple translations to mesh-based warps.

1 Simple Image Registration

Suppose that the template is $T(x, y)$ and the image that it is to be registered against is $I(x, y)$. Although we think of the template T as moving and deforming in the image I, it is easiest to evaluate the degree of match in the coordinate frame of the template. For each pixel in the template window, we determine the corresponding pixel in I, (bilinearly) interpolate the image there, and then difference the value obtained from the template value; that is we warp I back onto the template coordinate frame and do the difference there. Suppose that the template deformation model is:

$$(x, y) \ \rightarrow \ (\rho^x(x, y; \mathbf{p}), \rho^y(x, y; \mathbf{p}))\,. \tag{1}$$

The coordinates (x, y) in the template window are mapped onto $(\rho^x(x, y; \mathbf{p}), \rho^y(x, y; \mathbf{p}))$ in the image I where $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ is a vector of n parameters. In the simple case of a translation $\rho^x(x, y; \mathbf{p}) = x + p_1$ and $\rho^y(x, y; \mathbf{p}) = y + p_2$. The goal of image registration is then to minimize:

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

over the set of parameters $\mathbf{p}$, and where W is the template window.

Minimizing the expression in Equation (2) is a non-linear optimization problem over the unknown parameters $\mathbf{p}$. This optimization is typically performed using a form of gradient descent. In such algorithms, the best current estimate of the parameters is iteratively updated until it converges, hopefully to the globally optimal solution, although all we can really guarantee is convergence to a local minimum. Assume that $\mathbf{p} = (p_1, \ldots, p_n)$ contains the current best estimate of the parameters (known), and $\Delta\mathbf{p} = (\Delta p_1, \ldots, \Delta p_n)$ is the (unknown) additive update to $\mathbf{p}$.

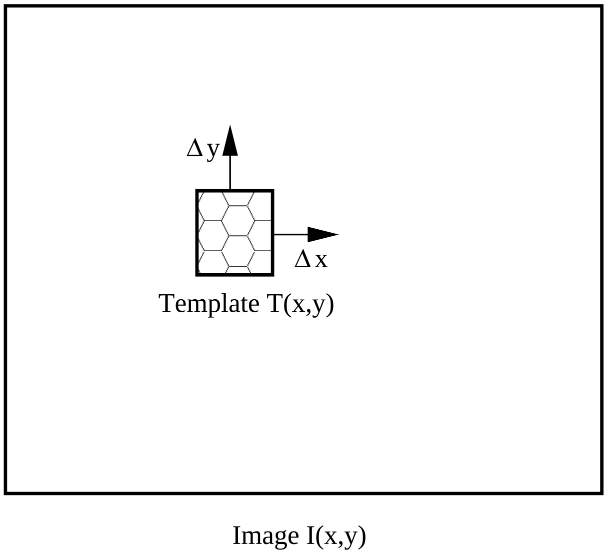

Figure 1: At the most basic level, image registration (or alignment) consists of moving (and deforming) a template to minimize the difference between the (deformed) template and an image. In many cases the template is (a cropped sub-region of) another image, but it could also be a complex model of image appearance, such as an Active Appearance Model (AAM) [6]. The way in which the template can move and deform depends upon the application in question and can vary from a simple translation, as in the Lucas-Kanade optical flow algorithm [7], to a complex mesh-based alignment, as is the case for AAMs. The "difference" between the template and the image is usually taken to be the "sum of squared differences" (SSD) but can be more complex and, for example, take into account illumination variation [4], or even more general appearance variations [6]. Image registration is normally performed using gradient descent on the registration parameters (see [8] for a general formulation of this step), but linear regression has also been proposed as a way to estimate a linear approximation to the parameter updates in the gradient descent algorithm [6].

Following [8] we now derive one formulation of how to estimate $\Delta\mathbf{p}$. The first step consists of performing a first order Taylor expansion on $I\left(\rho^x(x,y;\mathbf{p}+\Delta\mathbf{p}), \rho^y(x,y;\mathbf{p}+\Delta\mathbf{p})\right)$. Initially, we just perform a first order Taylor expansion on $\rho^x(x,y;\mathbf{p}+\Delta\mathbf{p})$:

$$\rho^x(x,y;\mathbf{p}+\Delta\mathbf{p}) \;\approx\; \rho^x(x,y;\mathbf{p}) + \sum_{i=1}^{n} \frac{\partial \rho^x}{\partial p_i}\Delta p_i. \tag{3}$$

The equivalent result holds for $\rho^y(x,y;\mathbf{p}+\Delta\mathbf{p})$. We now have an expression of the form $I(x+\Delta x, y+\Delta y)$ where x represents $\rho^x(x,y;\mathbf{p})$ and Δx represents $\sum_{i=1}^{n}\frac{\partial\rho^x}{\partial p_i}\Delta p_i$ (and similarly for y and Δy.) Performing a Taylor expansion on $I(x+\Delta x, y+\Delta y)$ gives:

$$I(x+\Delta x, y+\Delta y) \;\approx\; I(x,y) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y. \tag{4}$$

Combining Equations (3) and (4) gives: $I\left(\rho^x(x,y;\mathbf{p}+\Delta\mathbf{p}), \rho^y(x,y;\mathbf{p}+\Delta\mathbf{p})\right) \approx$

$$I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) \;+\; \sum_{i=1}^{n}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i}\right]\Delta p_i. \tag{5}$$

In this expression $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ are evaluated at $(\rho^x(x,y;\mathbf{p}),\rho^y(x,y;\mathbf{p}))$, and $\frac{\partial \rho^x}{\partial p_i}$ and $\frac{\partial \rho^y}{\partial p_i}$ are evaluated at $(x,y;\mathbf{p})$. Combining Equation (5) and Equation (2) gives the following expression to be minimized with respect to $\Delta\mathbf{p} = (\Delta p_1, \ldots, \Delta p_n)$:

$$\sum_{(x,y)\in W} \left[\sum_{i=1}^{n} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i} \right] \Delta p_i \; + \; \left(I\left(\rho^x(x,y;\mathbf{p}),\rho^y(x,y;\mathbf{p})\right) - T(x,y) \right) \right]^2 . \qquad (6)$$

If we set $E(x,y) = I\left(\rho^x(x,y;\mathbf{p}),\rho^y(x,y;\mathbf{p})\right) - T(x,y)$ to be the *error image* between the backwards warped version of $I(x,y)$ and the template $T(x,y)$, this equation simplifies to:

$$\sum_{(x,y)\in W} \left[\sum_{i=1}^{n} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i} \right] \Delta p_i \; + \; E(x,y) \right]^2 . \qquad (7)$$

The partial derivative of this expression with respect to Δp_i is:

$$2 \cdot \sum_{(x,y)\in W} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i} \right] \cdot \left[\sum_{j=1}^{n} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_j} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_j} \right] \Delta p_i \; + \; E(x,y) \right] . \qquad (8)$$

Setting this expression to zero and re-organizing gives:

$$\sum_{j=1}^{n} \sum_{(x,y)\in W} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i} \right] \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_j} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_j} \right] \Delta p_i = - \sum_{(x,y)\in W} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i} \right] E(x,y) \qquad (9)$$

Combining the equations for $i = 1, \ldots n$ into one vector equation gives:

$$A \cdot \begin{pmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_n \end{pmatrix} \; = \; -\mathbf{b} \qquad (10)$$

where A is the $n \times n$ matrix with (i,j) element $\sum_{(x,y)\in W} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i} \right] \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_j} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_j} \right]$ and $\mathbf{b}$ is the $1 \times n$ column vector with i^{th} element $\sum_{(x,y)\in W} \left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i} \right] E(x,y)$.

The gradient descent algorithm for minimizing the expression in Equation (2) therefore operates as follows. An initial estimate (guess) is made for the parameters $\mathbf{p}$. The updates to the parameters $\Delta\mathbf{p}$ are then estimated, as will be described in a moment. These updates are then added to $\mathbf{p}$ to give $\mathbf{p} + \Delta\mathbf{p}$. These steps are iterated until $\mathbf{p}$ converges to a steady value (i.e. until $\Delta\mathbf{p}$ becomes close enough to zero.) Finally, the updates $\Delta\mathbf{p}$ are estimated as follows:

1. Using the current estimate of $\mathbf{p}$, $I(x,y)$ is warped backwards onto the coordinate frame of $T(x,y)$. The template $T(x,y)$ is then subtracted from it to form the error image $E(x,y)$.
2. For each pixel $(x,y) \in W$ the template window, the values of $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial \rho^x}{\partial p_i}$, and $\frac{\partial \rho^y}{\partial p_i}$ are evaluated, the second pair at $(x,y;\mathbf{p})$, the first pair at $(\rho^x(x,y;\mathbf{p}),\rho^y(x,y;\mathbf{p}))$.
3. The results of Steps (1) and (2) are combined to estimate the matrix A and the vector b.
4. The linear system in Equation (10) is solved to give $\Delta\mathbf{p}$.

2 More General (Linear Subspace) Template Models

In the previous section, image alignment was formulated as minimizing the error from a single template image $T(x,y)$:

$$\sum_{(x,y)\in W} \left[I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y)\right]^2 = \left\| I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y)\right\|^2. \quad (11)$$

Often it is assumed that the template 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 [4], but could easily model more general appearance changes [2, 6]. Suppose that the vectors $A_1(x,y), \ldots, A_d(x,y)$ are an (orthonormal) basis for the linear subspace. The alignment problem is then posed as one of minimizing:

$$\left\| I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y) + \sum_{i=1}^{d} \lambda_i A_i(x,y) \right\|^2 \quad (12)$$

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 (12) can be rewritten as:

$$\begin{aligned} &\left\| I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y) + \textstyle\sum_{i=1}^{d} \lambda_i A_i(x,y) \right\|^2_{\mathrm{span}(A_i)^\perp} \quad + \\ &\left\| I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y) + \textstyle\sum_{i=1}^{d} \lambda_i A_i(x,y) \right\|^2_{\mathrm{span}(A_i)} \end{aligned} \quad (13)$$

where $\|\cdot\|^2_L$ denotes the (square of the) norm of the vector projected into the linear subspace L. The first of the two components in Equation (13) immediately simplifies. 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 wish to minimize:

$$\begin{aligned} &\left\| I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y) \right\|^2_{\mathrm{span}(A_i)^\perp} \quad + \\ &\left\| I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y) + \textstyle\sum_{i=1}^{d} \lambda_i A_i(x,y) \right\|^2_{\mathrm{span}(A_i)} \end{aligned} \quad (14)$$

over both $\mathbf{p}$ and λ_i. 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 of the expression in Equation(14) can be found sequentially by first minimizing:

$$\left\| I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y) \right\|^2_{\mathrm{span}(A_i)^\perp} \quad (15)$$

with respect to $\mathbf{p}$ alone, and then using that optimal value of $\mathbf{p}$ as a constant to minimize:

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

with respect to the λ_i. Assuming that the basis vectors A_i are othonormal, the second minimization is easy. The solution is simply:

$$\lambda_i \;=\; -\sum_{(x,y)\in W} A_i(x,y)\cdot\left[I\left(\rho^x(x,y;\mathbf{p}),\rho^y(x,y;\mathbf{p})\right) - T(x,y)\right] \;=\; -\sum_{(x,y)\in W} A_i(x,y)\cdot E(x,y) \tag{17}$$

where $E(x,y)$ is the final error image obtained after doing the first minimization (i.e. the minimization of Equation (15) with respect to $\mathbf{p}$.)

The only remaining step is then to minimize:

$$\left\| I\left(\rho^x(x,y;\mathbf{p}),\rho^y(x,y;\mathbf{p})\right) - T(x,y)\right\|^2_{\mathrm{span}(A_i)^\perp} \tag{18}$$

with respect to $\mathbf{p}$. The equivalent of Equation (7) is therefore:

$$\left\| \sum_{i=1}^{n}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i}\right]\Delta p_i \;+\; E(x,y)\right\|^2_{\mathrm{span}(A_i)^\perp}. \tag{19}$$

This is equivalent to:

$$\left\| \sum_{i=1}^{n}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i}\right]_{\mathrm{span}(A_i)^\perp}\Delta p_i \;+\; \left[E(x,y)\right]_{\mathrm{span}(A_i)^\perp}\right\|^2 \tag{20}$$

where $[\cdot]_{\mathrm{span}(A_i)^\perp}$ denotes the projection of a vector into the linear subspace $\mathrm{span}(A_i)^\perp$. We can then add in the norm of the component of $E(x,y)$ in the linear subspace $\mathrm{span}(A_i)$ since it is a constant. We therefore need to minimize:

$$\left\| \sum_{i=1}^{n}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i}\right]_{\mathrm{span}(A_i)^\perp}\Delta p_i \;+\; \left[E(x,y)\right]_{\mathrm{span}(A_i)^\perp}\right\|^2 + \left\|\left[E(x,y)\right]_{\mathrm{span}(A_i)}\right\|^2. \tag{21}$$

The two norm-squared terms can then be combined because the vectors they are norms of are orthogonal:

$$\left\| \sum_{i=1}^{n}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i}\right]_{\mathrm{span}(A_i)^\perp}\Delta p_i \;+\; \left[E(x,y)\right]_{\mathrm{span}(A_i)^\perp} + \left[E(x,y)\right]_{\mathrm{span}(A_i)}\right\|^2. \tag{22}$$

This then equals:

$$\left\| \sum_{i=1}^{n}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i}\right]_{\mathrm{span}(A_i)^\perp}\Delta p_i \;+\; E(x,y)\right\|^2. \tag{23}$$

The derivation of the update equations Equation (10) then follows exactly as before, but the definitions of A and $\mathbf{b}$ need to be modified slightly. The (i,j) element of A is:

$$\sum_{(x,y)\in W}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_i}\right]_{\mathrm{span}(A_i)^\perp}\left[\frac{\partial I}{\partial x}\frac{\partial \rho^x}{\partial p_j} + \frac{\partial I}{\partial y}\frac{\partial \rho^y}{\partial p_j}\right]_{\mathrm{span}(A_i)^\perp} \tag{24}$$

and the i^{th} element of $\mathbf{b}$ is:

$$\sum_{(x,y)\in W} \left[\frac{\partial I}{\partial x} \frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y} \frac{\partial \rho^y}{\partial p_i} \right]_{\text{span}(A_i)^\perp} E(x,y) \tag{25}$$

In summary, the image registration algorithm with linear appearance variation is exactly the same as without, except: (1) $\left[\frac{\partial I}{\partial x} \frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y} \frac{\partial \rho^y}{\partial p_i} \right]$ must be projected into the subspace orthogonal to $\text{span}(A_i)$ and (2) once the parameters $\mathbf{p}$ have been estimated, Equation (17) is used to estimate the appearance parameters λ_i. The projection into the subspace orthogonal to $\text{span}(A_i)$ could be written as a single (large) matrix product, but in practice it is computed by removing the component of $\left[\frac{\partial I}{\partial x} \frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y} \frac{\partial \rho^y}{\partial p_i} \right]$ in each of the orthogonal directions $A_1, \ldots, A_d$ in turn.

3 An Efficient Alignment Algorithm for Warp Groups

Up to now, the matrix A defined in Equation (24) and the vector $\mathbf{b}$ defined in Equation (25) must both be updated in each iteration. Suppose hypothetically for now that $\left[\frac{\partial I}{\partial x} \frac{\partial \rho^x}{\partial p_i} + \frac{\partial I}{\partial y} \frac{\partial \rho^y}{\partial p_i} \right]_{\text{span}(A_i)^\perp}$ is a constant. This means that the matrix A is a constant. The algorithm can then be made far more efficient. If the matix A is, in fact, constant, it can be inverted once and for all and combined with Equation (10) to give a simple expression for the parameter updates:

$$\begin{pmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_n \end{pmatrix} = \sum_{(x,y)\in W} A^{-1} \begin{pmatrix} \left[\frac{\partial I}{\partial x} \frac{\partial \rho^x}{\partial p_1} + \frac{\partial I}{\partial y} \frac{\partial \rho^y}{\partial p_1} \right]_{\text{span}(A_i)^\perp} \\ \left[\frac{\partial I}{\partial x} \frac{\partial \rho^x}{\partial p_2} + \frac{\partial I}{\partial y} \frac{\partial \rho^y}{\partial p_2} \right]_{\text{span}(A_i)^\perp} \\ \vdots \\ \left[\frac{\partial I}{\partial x} \frac{\partial \rho^x}{\partial p_n} + \frac{\partial I}{\partial y} \frac{\partial \rho^y}{\partial p_n} \right]_{\text{span}(A_i)^\perp} \end{pmatrix} \cdot E(x,y) \tag{26}$$

If everything on the right hand side of this equation is constant except for $E(x,y)$, then A^{-1} and the other (large) matrix can be multipled out offline, separately for each pixel $(x,y) \in W$, to give an image that when multiplied by $E(x,y)$ and the results added up (i.e. dot-producted) directly gives the parameter updates $\Delta\mathbf{p}$. The algorithm to estimate $\Delta\mathbf{p}$ then consists of Step (1) as above to estimate E followed by a simple dot-product with a constant template-size image to directly estimate $\Delta\mathbf{p}$. (There will be one such dot-product for each parmeter; i.e. there will be n in total.)

Unfortunately, however, the components of A, $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial \rho^x}{\partial p_i}$, and $\frac{\partial \rho^y}{\partial p_i}$ are unlikely to be constant. A particular problem are the first pair of these. These are samples of the gradient of I. The location of these samples will vary with $\mathbf{p}$ and so their values are exceedingly unlikely to be constant. It is a shame that these values are from the gradient of I rather than T. Since the template does not move, the samples of its gradient are constant. The key then is to try to repeat the derivation, but reversing the roles of I and T. This is what was tried in [4]. The approach goes back to the orginal formulation of the problem:

$$\sum_{(x,y)\in W} \left[I\left(\rho^x(x,y;\mathbf{p}), \rho^y(x,y;\mathbf{p})\right) - T(x,y) \right]^2 \tag{27}$$

Reversing the roles of I and T means applying a coordinate transformation like:

$$(\overline{x}, \overline{y}) \;=\; (\rho^x(x, y; \mathbf{p}), \rho^y(x, y; \mathbf{p})) \tag{28}$$

or equivalently its inverse:

$$(x, y) \;=\; \left(\overline{\rho}^{\overline{x}}(\overline{x}, \overline{y}; \mathbf{p}), \overline{\rho}^{\overline{y}}(\overline{x}, \overline{y}; \mathbf{p})\right) \tag{29}$$

At first glance, it might be thought that the result of transforming Equation (27) is:

$$\sum_{(x,y) \in W} \left[I\left(\overline{x}, \overline{y}\right) - T\left(\overline{\rho}^{\overline{x}}(\overline{x}, \overline{y}; \mathbf{p}), \overline{\rho}^{\overline{y}}(\overline{x}, \overline{y}; \mathbf{p})\right)\right]^2 \tag{30}$$

but this is incorrect. Really the summation is an approximation to an integral and the determinant of the Jacobian of the change of coordinates must be included. The correct expression is therefore:

$$\sum_{(x,y) \in W} \left| \frac{\partial(\overline{\rho}^{\overline{x}}, \overline{\rho}^{\overline{y}})}{\partial(\overline{x}, \overline{y})} \right| \left[I\left(\overline{x}, \overline{y}\right) - T\left(\overline{\rho}^{\overline{x}}(\overline{x}, \overline{y}; \mathbf{p}), \overline{\rho}^{\overline{y}}(\overline{x}, \overline{y}; \mathbf{p})\right)\right]^2 \tag{31}$$

The symmetry between I and T is therefore not exact and the roles of the two images cannot simply be interchanged. This was pointed out in [4]. Hager and Belhumeur then proceeded to analyze the determinant of the Jacobian:

$$\left| \frac{\partial(\overline{\rho}^{\overline{x}}, \overline{\rho}^{\overline{y}})}{\partial(\overline{x}, \overline{y})} \right| \tag{32}$$

and show that *if it takes a particularly simple form* (can be factored into a matrix that depends only upon $\mathbf{p}$ multiplied by a matrix that depends only upon (x, y)) there is a way to obtain the efficiency improvements described above. (Specifically, the term that depens upon $\mathbf{p}$ can be moved in front of the summation sign and then dealt with at the end of the computation. The term that only depends upon (x, y) is constant across iterations and so can be included in the pre-computation.)

The approach described in [4], however, only applies to certain transformations including translations, rotations and scalings, and affine warps. It cannot be used for homographies, for example, as well as many more complicated transformations such as those described in [8].

References

[1] 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.

[2] 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.

[3] 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):322–336, 2000.

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

[5] M. Irani and S. Peleg. Improving resolution by image restoration. *Computer Vision, Graphics, and Image Processing*, 53:231–239, 1991.

[6] 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.

[7] 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.

[8] 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.