Efficiently Estimating Projective Transformations

Richard Radke, Peter Ramadge*
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544
(609) 258-4645
{rjradke,ramadge}@ee.princeton.edu

Tomio Echigo
IBM Research
Tokyo Research Laboratory
1623-14 Shimotsuruma
Yamato-shi, Kanagawa, Japan
echigo@trl.ibm.co.jp

June 27, 2001

Abstract

Projective transformations relate the coordinates of images that are taken by either a camera that undergoes only rotation while imaging an arbitrary scene, or one that rotates and translates while imaging a planar surface. Estimating the eight parameters of a projective transformation between a pair of image planes induces a global, dense correspondence between them that can be used for registration or mosaicking. This is a standard problem in image processing and computer vision.

The projective transformation estimation problem is typically posed as the minimization of a nonlinear functional of eight parameters, and solved with an "off-the-shelf" numerical algorithm. Here it is shown that in fact, this minimization can be analytically reduced to a nonlinear problem in only two parameters. Any descent algorithm to solve the eight-dimensional minimization can be modified to produce an algorithm for the two-dimensional problem. Several algorithms based on the two-dimensional problem are proposed and results are presented on data from real images to experimentally verify their superiority. Not only are the proposed algorithms efficient, but they are also robust to the types of measurement noise that could be introduced by poorly matched corresponding points or outliers in the data sets.

Index terms: projective transformations, homographies, collineations, mosaicking, registration

*This research was partially supported by grants from the IBM Tokyo Research Laboratory and the New Jersey Center for Multimedia Research. Peter Ramadge is the corresponding author.

1 Introduction

A projective transformation maps a point $w \in \mathbb{R}^2$ to $w' \in \mathbb{R}^2$ by:

$$w' = \frac{Aw + b}{c^T w + d} \tag{1}$$

where $A \in \mathbb{R}^{2\times 2}$, $b, c \in \mathbb{R}^2$, and $d \in \mathbb{R}$ are such that $\left|\begin{smallmatrix} A & b \\ c^T & d \end{smallmatrix}\right| \neq 0$. An affine transformation is a special case of a projective transformation with $c = 0$ and $d = 1$.

The estimation of the parameters of a two-dimensional projective transformation is a standard problem that arises in image and video processing. Typical applications of the estimated parameters are the synthesis of an image mosaic from a set of images of a static scene taken by a rotating camera [1, 2, 3, 4, 5], and the registration of images of a planar surface taken by multiple separated cameras [6, 7].

We can pose the projective transformation estimation problem as a least-squares minimization based on a finite set of noisy point samples of the underlying transformation. This results in an eight-dimensional nonlinear nonquadratic minimization problem. Such a problem is typically solved numerically using an 'off-the-shelf' procedure such as the Gauss-Newton or Levenberg-Marquardt algorithm [8]. In special cases, approximations are possible that enable closed-form solutions. For example, when the c parameters are very close to 0, Tan [9] introduced an approximation to make the least-squares problem linear. Kanatani [10] proposed a tensor-based approximation that reduces the estimation problem to an eigendecomposition. However, the mapping (1) is very sensitive to changes in the c parameters, and as these parameters deviate from 0, the above approximations quickly diverge from the solution to the nonlinear problem. Instead of using a set of point correspondences as a basis for estimating a projective transformation, Mann and Picard [2] proposed an iterative technique for simultaneously estimating the transformation parameters and optical flow over an entire image pair. They used bilinear approximations to the projective

transformations in each step in order to simplify the estimation.

The main contribution of this paper is an efficient and robust new framework for solving the projective transformation estimation problem. We show in Section 4 that the general least-squares problem for estimating a projective transformation can be analytically reduced to a two-dimensional nonquadratic minimization problem. Some properties of the two-dimensional cost function are discussed in Section 5. In Sections 6 and 7 we discuss issues involved with the practical minimization of the cost function by analyzing its gradient and Hessian, and show that in theory, any descent algorithm for the eight-dimensional problem can be modified to produce a more effective descent algorithm for the two-dimensional problem. We provide experimental results in Section 8 to show that Newton methods based on the two-dimensional problem outperform analogous methods applied to the eight-dimensional problem. Furthermore, we propose an approximate second-derivative method that is quite robust to measurement noise. A brief summary of some of our results originally appeared in [11].

2 Preliminaries

We consider the situation where the same static scene is imaged by two cameras $\mathcal{C}_0$ and $\mathcal{C}_1$. These could be two physically separate cameras, or a single moving camera at different points in time. The images are formed on coordinatized image planes $\mathcal{P}_0$ and $\mathcal{P}_1$, respectively. Let the scene coordinates of a point P in the $\mathcal{C}_0$ coordinate system be (X, Y, Z), and in the $\mathcal{C}_1$ coordinate system be (X', Y', Z'). We denote the corresponding image coordinates of P in $\mathcal{P}_0$ and $\mathcal{P}_1$ by $w = (x, y)$ and $w' = (x', y')$, respectively. We assume idealized pinhole cameras that produce images by perspective projection. Thus,

$$\begin{aligned} x &= f_0 \tfrac{X}{Z} \qquad & y &= f_0 \tfrac{Y}{Z} \\ x' &= f_1 \tfrac{X'}{Z'} \qquad & y' &= f_1 \tfrac{Y'}{Z'} \end{aligned} \tag{2}$$

Here f_0 and f_1 are the focal lengths of $\mathcal{C}_0$ and $\mathcal{C}_1$, respectively. We assume that the two cameras

are related by a rigid motion. Hence, the $\mathcal{C}_1$ coordinate system can be expressed as a rotation R of the $\mathcal{C}_0$ coordinate system followed by a translation $[t_X\ t_Y\ t_Z]^T$. That is,

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix} \tag{3}$$

By substituting (3) into (2), we obtain:

$$x' = \frac{r_{11}\frac{f_1}{f_0}x + r_{12}\frac{f_1}{f_0}y + r_{13}f_1 + \frac{t_X f_1}{Z}}{\frac{r_{31}}{f_0}x + \frac{r_{32}}{f_0}y + r_{33} + \frac{t_Z}{Z}} \tag{4}$$

$$y' = \frac{r_{21}\frac{f_1}{f_0}x + r_{22}\frac{f_1}{f_0}y + r_{23}f_1 + \frac{t_Y f_1}{Z}}{\frac{r_{31}}{f_0}x + \frac{r_{32}}{f_0}y + r_{33} + \frac{t_Z}{Z}} \tag{5}$$

Here the r_{ij} are the elements of the rotation matrix R given in (3).

We note two special cases of (4) and (5). In the first case, $t_X = t_Y = t_Z = 0$, corresponding to a camera whose optical center undergoes no translation, we obtain

$$x' = \frac{r_{11}\frac{f_1}{f_0}x + r_{12}\frac{f_1}{f_0}y + r_{13}f_1}{\frac{r_{31}}{f_0}x + \frac{r_{32}}{f_0}y + r_{33}}$$

$$y' = \frac{r_{21}\frac{f_1}{f_0}x + r_{22}\frac{f_1}{f_0}y + r_{23}f_1}{\frac{r_{31}}{f_0}x + \frac{r_{32}}{f_0}y + r_{33}}$$

In the second case, we assume that all scene points satisfy $k_1X + k_2Y + k_3Z = 1$, corresponding to a planar surface. Then (4)-(5) become:

$$x' = \frac{(r_{11}\frac{f_1}{f_0} + t_X f_1 k_1)x + (r_{12}\frac{f_1}{f_0} + t_X f_1 k_2)y + (r_{13}f_1 + t_X f_1 k_3)}{(\frac{r_{31}}{f_0} + t_Z k_1)x + (\frac{r_{32}}{f_0} + t_Z k_2)y + (r_{33} + t_Z k_3)}$$

$$y' = \frac{(r_{21}\frac{f_1}{f_0} + t_Y f_1 k_1)x + (r_{22}\frac{f_1}{f_0} + t_Y f_1 k_2)y + (r_{23}f_1 + t_Y f_1 k_3)}{(\frac{r_{31}}{f_0} + t_Z k_1)x + (\frac{r_{32}}{f_0} + t_Z k_2)y + (r_{33} + t_Z k_3)}$$

In each case, we have reduced the relationship between image coordinates to the form of a projective transformation.

3 Projective Transformations

For $M \in \mathbb{R}^{3\times 3}$ with $\det(M) \neq 0$, i.e., $M \in GL(3)$, write $M = \left(\begin{smallmatrix} A & b \\ c^T & d \end{smallmatrix}\right)$ with $A \in \mathbb{R}^{2\times 2}$, b and $c \in \mathbb{R}^{2\times 1}$, and $d \in \mathbb{R}$. Then the transformation g_M of the plane defined by

$$g_M(w) = \frac{Aw + b}{c^T w + d} \tag{6}$$

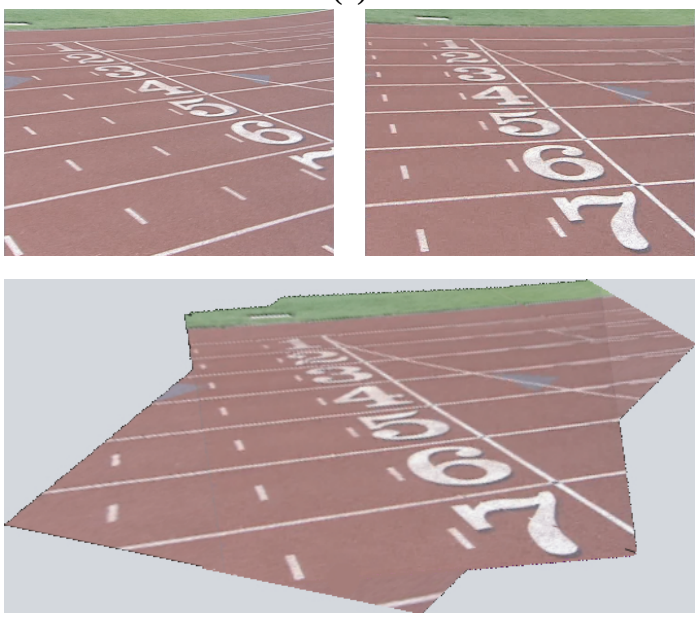
is called a projective transformation (or collineation or homography) with homogeneous coordinates M.

The set of projective transformations of the plane forms a group $\mathcal{G}$ under function composition. In the remainder of the development, we will normalize $d = 1$, so that a projective transformation is uniquely characterized by eight parameters $M = (A, b, c)$. This excludes the set of transformations with $d = 0$; however, this subset of transformations is not usually of interest. The matrix A reflects changes in rotation and scale of the w plane, and the vector b reflects translational offset. The two "projective" parameters of c account for the keystoning effects of perspective projection.

Projective transformations relate the image coordinates of a pair of images taken by cameras whose optical center undergoes no translation, or of images of a planar surface. Hence, projective transformations can be used to align several such images in the same frame of reference, as illustrated in Figure 1. The nonlinear warping of the images is clearly visible.

Note that g_M in (6) is defined at all points of $\mathbb{R}^2$ except those w on the line $c^T w + 1 = 0$, which is called the singular line of the fixed transformation g_M. Along this line $Aw + b \neq 0$, since $M \in GL(3)$. In the two special cases above, singular lines have a geometric interpretation. The singular line is the intersection of the image plane $\mathcal{P}_0$ with the plane $Z' = 0$ corresponding to the parallel transport of the image plane $\mathcal{P}_1$ to the center of projection O_1. This is illustrated in Figure 2. In practical situations (e.g. when the image planes are of finite extent, and one camera is not in the field of view of the other) all the points in $\mathcal{P}_0$, including the center or origin, lie to the side of

(a)



(b)

Figure 1: Applications of projective transformations. (a) Mosaicking images from a non-translating camera, (b) registering images of a planar surface taken by a translating camera.

the singular line with $c^T w + 1 > 0$.

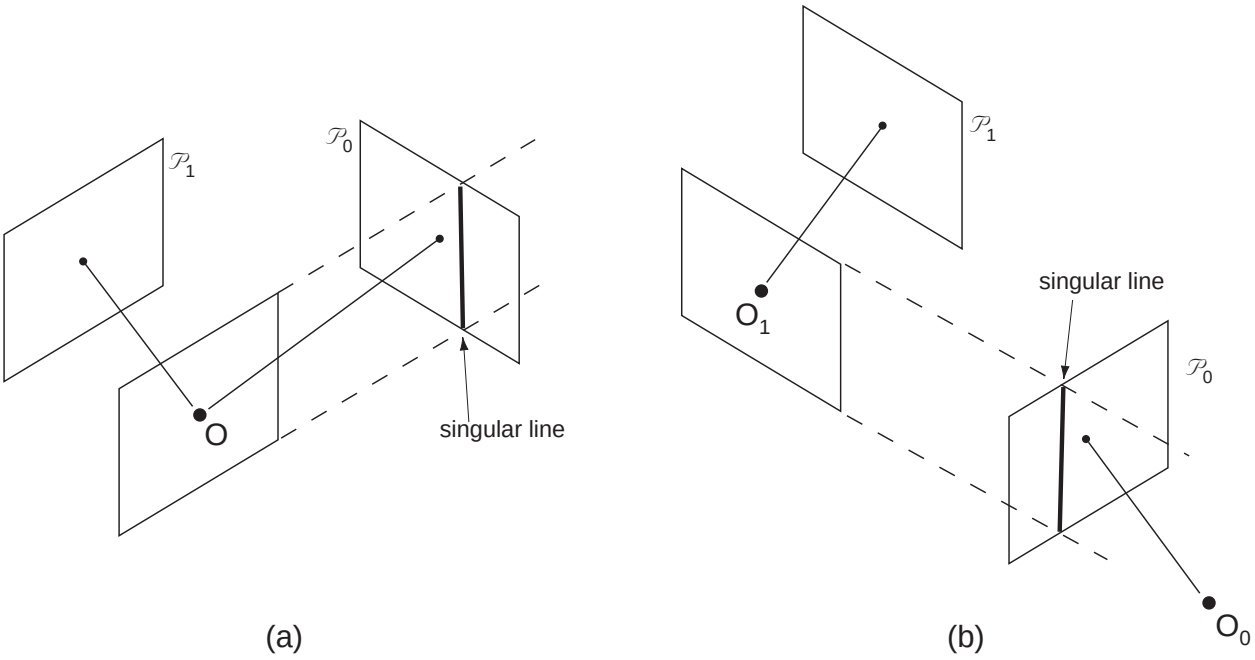


Figure 2: (a) Singular line in fixed-center case, (b) Singular line in translated case.

For a fixed projective transformation, hence for a fixed c, there is a line of w in $\mathcal{P}_0$ that lie on the corresponding singular line. Conversely, for a fixed $w \in \mathcal{P}_0$, there is a singular line of c in $\mathbb{R}^2$and a corresponding half-plane $c^T w + 1 > 0$. Since we will estimate a projective transformation from a set of fixed data points, the intersection of these half-planes in c-space form a set of special interest.

4 The Least-squares Estimate

We will consider a parameter estimation problem in which the data is a set of noisy measurements of matching points between a pair of images of a static scene. In practice, the noisy point samples originate from automatically generated or manually selected feature correspondences in an image pair such as similar blocks of pixels, intersections of lines, or corners. Our objective is to select the parameters $M = (A, b, c)$ so that g_M best fits a given set of point matches: $\{w_j \mapsto w'_j \in \mathbb{R}^2, j =$

$1, \ldots, N\}$. A case of special interest arises when the data consists of noisy samples of a fixed but unknown projective transformation g_{M^*}: $w'_j = g_{M^*}(w_j) + e_j,\ j = 1 \ldots N$. Here $e_j \in \mathbb{R}^2$ is the error in the measurement of $g_{M^*}(w_j)$. In this case we seek an estimate M of M^*.

An estimate M is, by definition, *admissible* if the singular line of g_M does not intersect the convex hull W of 0 and w_j, $j = 1, \ldots, N$. Since $0 \in W$, M is admissible if and only if $c^T w + 1 > 0$ for all $w \in W$. This is equivalent to the requirement that $c^T w_j + 1 > 0$, $j = 1, \ldots, N$. This defines an open convex set $C_o \subset \mathbb{R}^2$ of allowed values for c, and M is admissible if and only if $c \in C_o$. The set of admissible estimates is the open set $\{(A, b, c)\colon A \in \mathbb{R}^{2\times 2}, b \in \mathbb{R}^2, c \in C_o\}$. Note that admissibility does not require $M \in GL(3)$. Figure 3 illustrates the admissible region C_o generated by data points from an actual image pair.

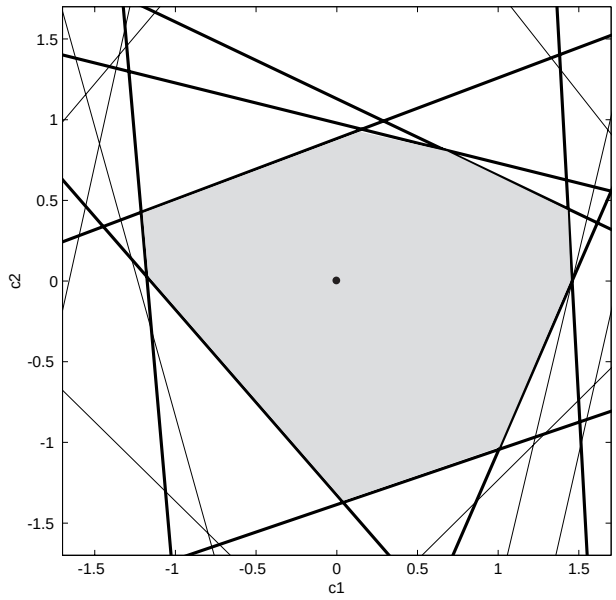


Figure 3: The admissible region C_o of the (c_1, c_2) plane generated by data points from actual images (see Figure 4). Thin lines represent singular lines; thick lines are singular lines that actively bound the admissible region.

The least-squares estimate $\hat{M} = (\hat{A}, \hat{b}, \hat{c})$ consists of those values of A, b and c that globally

minimize:

$$Q(M) = \frac{1}{2}\sum_{j=1}^{N}\left(w_j' - \frac{Aw_j + b}{c^T w_j + 1}\right)^T \left(w_j' - \frac{Aw_j + b}{c^T w_j + 1}\right) \tag{7}$$

over all admissible $M = (A, b, c)$. In general this estimate need not be an element of $GL(3)$ and hence need not itself be a projective transformation. However, for a wide range of reasonable models for the noise terms e_j, $j = 1, \ldots, N$, $\hat{M}$ will generically be an element of the open set $GL(3)$. We defer the proof that Q has a global minimum within the set of admissible estimates to the end of Section 6.

For a fixed data set, obtaining the least-squares estimate requires solving a nonlinear minimization problem over an open subset of 8-dimensional Euclidean space. However, as Theorem 1 below shows, the solution can also be obtained by solving a nonlinear minimization problem over an open convex subset of $\mathbb{R}^2$.

Theorem 1 *Assuming that the points w_j, $j = 1, \ldots, N$ are not colinear, the least-squares estimate $\hat{M}$ has the form $(A(\hat{c}), b(\hat{c}), \hat{c})$ and thus lies on the 2-dimensional submanifold $\mathcal{M} \triangleq \{(A, b, c)\colon A = A(c), b = b(c), c \in C_o\}$ of the eight dimensional space $\mathbb{R}^{2\times 2} \times \mathbb{R}^2 \times C_o$.*

Proof: Since $\hat{M}$ minimizes (7), it follows that we must have $D_A Q(\hat{M}) = 0$, $D_b Q(\hat{M}) = 0$, and $D_c Q(\hat{M}) = 0$. This yields the normal equations:

$$\hat{A}\sum_j \frac{w_j w_j^T}{(\hat{c}^T w_j + 1)^2} + \hat{b}\sum_j \frac{w_j^T}{(\hat{c}^T w_j + 1)^2} - \sum_j \frac{w_j' w_j^T}{\hat{c}^T w_j + 1} = 0 \tag{8}$$

$$\hat{A}\sum_j \frac{w_j}{(\hat{c}^T w_j + 1)^2} + \hat{b}\sum_j \frac{1}{(\hat{c}^T w_j + 1)^2} - \sum_j \frac{w_j'}{\hat{c}^T w_j + 1} = 0 \tag{9}$$

$$\sum_j \left(w_j' - \left(\frac{\hat{A}w_j + \hat{b}}{\hat{c}^T w_j + 1}\right)\right)^T \left(\frac{\hat{A}w_j + \hat{b}}{\hat{c}^T w_j + 1}\right)\frac{w_j}{\hat{c}^T w_j + 1} = 0 \tag{10}$$

We can rewrite (8) and (9) as a linear system:

$$[\ \hat{A}\quad \hat{b}\ ]W(\hat{c}) = V(\hat{c}) \tag{11}$$

where $W(c) \in \mathbb{R}^{3\times 3}$, $V(c) \in \mathbb{R}^{2\times 3}$ are the functions of $c \in \mathbb{R}^2$ and the data points given by:

$$W(c) \;=\; \begin{bmatrix} \sum_{j=1}^N \frac{w_j w_j^T}{q_j^2(c)} & \sum_{j=1}^N \frac{w_j}{q_j^2(c)} \\ \\ \sum_{j=1}^N \frac{w_j^T}{q_j^2(c)} & \sum_{j=1}^N \frac{1}{q_j^2(c)} \end{bmatrix} \tag{12}$$

$$V(c) \;=\; \begin{bmatrix} \sum_{j=1}^N \frac{w_j' w_j^T}{q_j(c)} & \sum_{j=1}^N \frac{w_j'}{q_j(c)} \end{bmatrix} \tag{13}$$

Here $q_j(c) = c^T w_j + 1$. Therefore, defining

$$[\ A(c)\quad b(c)\ ] = V(c)W^{-1}(c) \tag{14}$$

we have $(\hat{A}, \hat{b}, \hat{c}) = (A(\hat{c}), b(\hat{c}), \hat{c})$ and the theorem follows. ∎

We make a standing assumption that the points $\{w_j : j = 1, \ldots, N\}$ are not all colinear in $\mathbb{R}^2$. This ensures that $W(c)$ is positive definite and hence that $A(c)$ and $b(c)$ are defined for all $c \in C_o$. To see this, consider an arbitrary vector in $\mathbb{R}^3$ partitioned as $\binom{v_1}{v_2}$ where $v_1 \in \mathbb{R}^2$ and $v_2 \in \mathbb{R}$. Then straightforward algebra shows that

$$\binom{v_1}{v_2}^T \begin{bmatrix} \sum_{j=1}^N \frac{w_j w_j^T}{q_j^2(c)} & \sum_{j=1}^N \frac{w_j}{q_j^2(c)} \\ \sum_{j=1}^N \frac{w_j^T}{q_j^2(c)} & \sum_{j=1}^N \frac{1}{q_j^2(c)} \end{bmatrix} \binom{v_1}{v_2}^T = \sum_{j=1}^N \frac{(v_1^T w_j + v_2)^2}{q_j^2} \geq 0$$

Thus $W(c)$ is positive definite if and only if for no nonzero $(v_1, v_2) \in \mathbb{R}^2 \times \mathbb{R}$ is it the case that $v_1^T w_j + v_2 = 0$ for all $w_j, j = 1, \ldots, N$. This is equivalent to the condition that $(w_j - w_k)^T v_1 = 0$ for all $j, k = 1, \ldots, N$, or that all the w_j are colinear.

In view of Theorem 1, we can define a two-dimensional cost functional $J : C_o \to \mathbb{R}$ by

$$J(c) = \frac{1}{2}\sum_{j=1}^{N}\left(w_j' - \frac{A(c)w_j + b(c)}{c^T w_j + 1}\right)^T \left(w_j' - \frac{A(c)w_j + b(c)}{c^T w_j + 1}\right) \tag{15}$$

$J(c)$ is simply the least-squares cost function restricted to the manifold $\mathcal{M}$. By construction, for any $M_o = (A(c_o), b(c_o), c_o) \in \mathcal{M}$, $Q(M_o) = J(c_o)$. Hence the global minimizing solution of $J(c)$ within C_o is $\hat{c}$. This reduces the determination of the least-squares estimate $\hat{M}$ to the minimization of J over C_o.

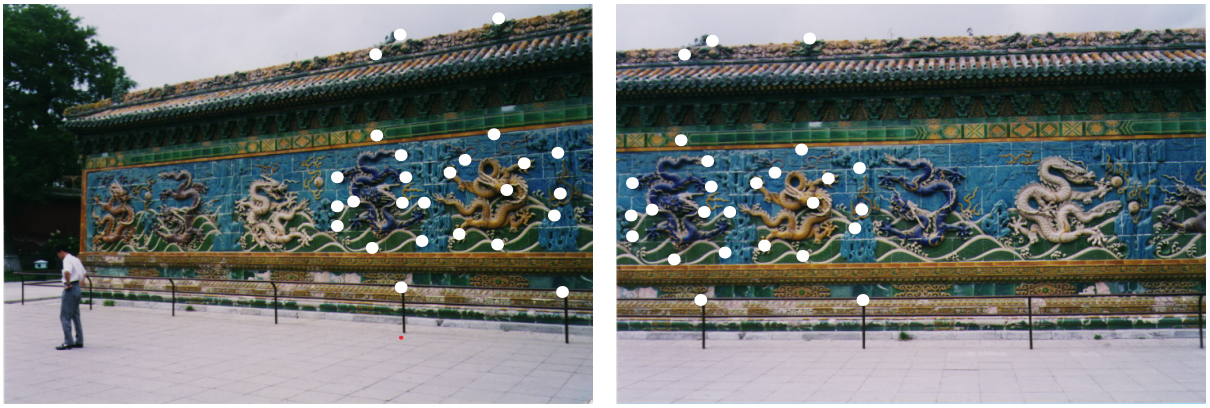
From the proof of the theorem, we can see that the 8-dimensional minimization of $Q(M)$ decouples into a nonlinear 2-dimensional minimization of c and a solution of a linear system for the "affine" parameters (A, b). This can be viewed as a specific case of a general mixed least-squares problem considered by Golub and Pereyra [12] that separates into linear and nonlinear variables. We go into considerably more detail here, exploring the structure of our specific problem.

Casting the problem in a two-dimensional setting allows us to visualize the cost function and the steps that a minimization algorithm takes. We shall show in Section 6 that in addition to being of reduced dimensionality, the cost function $J(c)$ can be numerically minimized more efficiently than the cost function $Q(M)$.

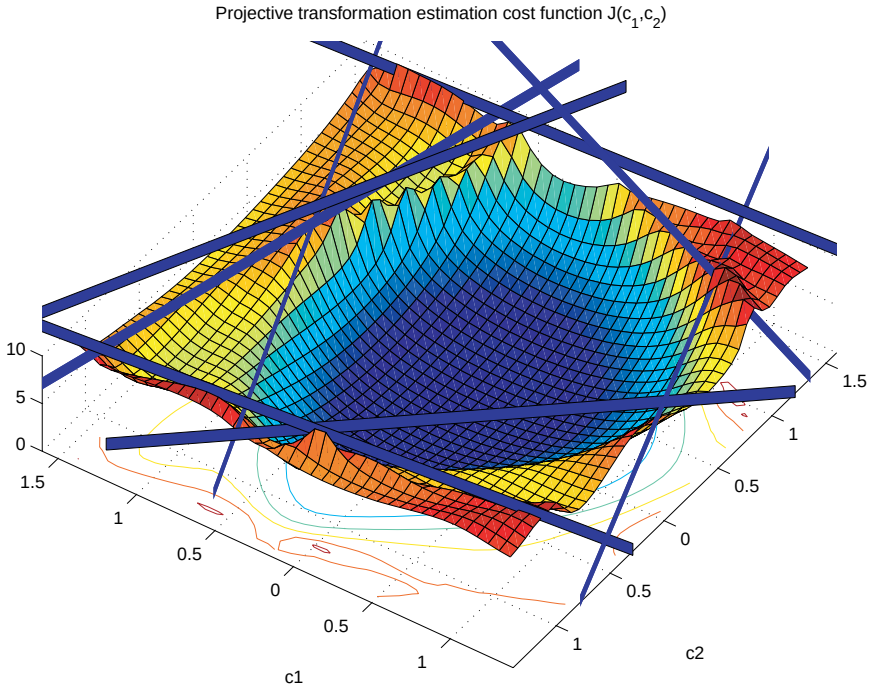
Figures 4a and 4b illustrate a pair of natural images, and a set of 25 noisy matching points between the images. Figure 4c illustrates the cost function J graphed over the region C_o for this set of data points. As can be seen, the cost function J has a single minimum within C_o, located at the bottom of a deep bowl.

5 The Behavior of J on Singular Lines

For c^* on one or more singular lines, the matrices $W(c^*)$ and $V(c^*)$ that define $(A(c^*), b(c^*))$ in (14) are not defined. However, below we provide two results concerning the finiteness and



(a) (b)



(c)

Figure 4: (a) and (b) A pair of natural images taken by a rotating camera with 25 matching points indicated. (c) The cost function $J(c_1, c_2)$ corresponding to the set of matched data points. The thick lines are the singular lines that actively bound C_o.

continuity of the functions $A(c)$ and $b(c)$ as c approaches a singular line. In the first theorem, we consider the behavior as we approach a point that lies on exactly one singular line.

Theorem 2 *Fix c^* such that $c^{*T}w_1 + 1 = 0$, and $c^{*T}w_j + 1 \neq 0$ for $j \neq 1$.*

1. *The limiting values $\lim_{c\to c^*}[\ A(c) \quad b(c)\ ]$ are well-defined, finite, and given by:*

$$[\ A_o \quad b_o\ ] \ = \ V_2 W_2^{-1}\left[I - \frac{pp^T W_2^{-1}}{p^T W_2^{-1} p}\right] \tag{16}$$

where $p = [w_1^T\ 1]^T$ and

$$W_2 \ = \ \begin{bmatrix} \sum_{j=2}^N \frac{w_j w_j^T}{q_j^2(c^*)} & \sum_{j=2}^N \frac{w_j}{q_j^2(c^*)} \\ \sum_{j=2}^N \frac{w_j^T}{q_j^2(c^*)} & \sum_{j=2}^N \frac{1}{q_j^2(c^*)} \end{bmatrix}$$

$$V_2 \ = \ \begin{bmatrix} \sum_{j=2}^N \frac{w_j' w_j^T}{q_j(c^*)} & \sum_{j=2}^N \frac{w_j'}{q_j(c^*)} \end{bmatrix}$$

2. *(A_o, b_o) is the solution to the well-defined constrained least-squares problem*

$$\begin{aligned} \min_{A,b} \quad & \frac{1}{2}\sum_{j=2}^N \left(w_j' - \frac{Aw_j + b}{c^{*T}w_j + 1}\right)^T \left(w_j' - \frac{Aw_j + b}{c^{*T}w_j + 1}\right) \\ s.t. \quad & Aw_1 + b = 0 \end{aligned}$$

3. $\lim_{c\to c^*} \frac{A(c)w_1 + b(c)}{c^T w_1 + 1} = w_1'$.

Proof: The proof can be found in the appendix. ∎

This result shows that, unlike the cost function $Q(M)$, the cost function $J(c)$ is finite and continuous along the singular lines. However, along singular lines the resulting least-squares projective transformation estimates are not members of $GL(3)$. The second and third parts of the theorem give precise characterizations of the limiting solution $(A(c^*), b(c^*))$ on a singular line.

Not only does the solution $(A(c^*), b(c^*))$ keep the cost function finite, but it zeros the offending data point's contribution to the cost function.

It is also important to consider the limiting behavior of $(A(c), b(c))$ as c approaches an intersection of two singular lines. To this end, we state the following theorem; the omitted proof is straightforward but tedious, and follows the same pattern as the proof of Theorem 2.

Theorem 3 *Fix c^* such that $c^{*T}w_1 + 1 = 0$ and $c^{*T}w_2 + 1 = 0$, with $w_1 \neq w_2$ and $c^{*T}w_j + 1 \neq 0$ for $j > 2$.*

1. *The limit $\lim_{c\to c^*}[\ A(c) \quad b(c)\ ]$ is well-defined, finite, and given by the solution of the following constrained minimization problem over $N-2$ data points:*

$$\begin{aligned} \min_{A,b} \quad & \frac{1}{2}\sum_{j=3}^{N}\left(w_j' - \frac{Aw_j + b}{c^{*T}w_j + 1}\right)^T\left(w_j' - \frac{Aw_j + b}{c^{*T}w_j + 1}\right) \\ s.t. \quad & Aw_1 + b = 0 \qquad (17) \\ & Aw_2 + b = 0 \qquad (18) \end{aligned}$$

2. *Moreover, $\lim_{c\to c^*}\frac{A(c)w_1+b(c)}{c^Tw_1+1} = w_1'$ and $\lim_{c\to c^*}\frac{A(c)w_2+b(c)}{c^Tw_2+1} = w_2'$.*

The intersection of three singular lines requires that three data points be colinear. In such an event one can prove a corresponding result on the finiteness and continuity of $A(c)$ and $b(c)$, and so on. At this point, we can prove the existence of global minima of J and Q:

Theorem 4 *If the closure of the set of admissible estimates $\bar{C}_o$ is compact and each vertex of the $\bar{C}_o$ polygon is determined by the intersection of at most two singular lines, then J has a global minimum in $\bar{C}_o$ and Q has a global minimum in $\mathbb{R}^{2\times 2} \times \mathbb{R}^2 \times \bar{C}_o$.*

Proof. From Theorems 2 and 3, we have that J is continuous over the compact set $\bar{C}_o$, so it must have a global minimizer $\hat{c}$ in $\bar{C}_o$. By Theorem 1, $Q(M)$ is globally minimized by $(A(\hat{c}), b(\hat{c}), \hat{c})$. ∎

6 Line-Search Descent

Typical algorithms for the minimization of a nonlinear function such as (7) operate in an iterative fashion as follows. Given a current approximation M_k of $\hat{M}$ select a direction d_k and take a step along the line from M_k in the direction d_k so as to decrease the objective function. The next approximation M_{k+1} is the value of M at this point. Typically the direction d_k is related to the gradient of the objective function evaluated at M_k.

When applied to the cost function $Q(M)$ such a scheme operates as follows. Let $M_k = (A_k, b_k, c_k)$, $k \geq 0$, be the approximation of $\hat{M}$ after step k and let $d_k = (F_k, g_k, h_k)$ denote the search direction used at step k. Then $(A_{k+1}, b_{k+1}, c_{k+1}) = (A_k, b_k, c_k) + \alpha_k(F_k, g_k, h_k)$, where the step size $\alpha_k \geq 0$ is selected to ensure that $Q(M_{k+1}) \leq Q(M_k)$.

For all such schemes we can make several observations. Let $M_o = (A_o, b_o, c_o)$ with $A_o \in \mathbb{R}^{2\times 2}$, and $b_o, c_o \in \mathbb{R}^2$. Define the projection of M_o onto $\mathcal{M}$ to be $P(M_o) \stackrel{\Delta}{=} (A(c_o), b(c_o), c_o)$.

Theorem 5 *Let $d = (F, g, h)$ with $F \in \mathbb{R}^{2\times 2}$, and $g, h \in \mathbb{R}^2$. Then*

1. *For any M_o, $J(c_o) = Q(P(M_o)) \leq Q(M_o)$.*

2. *For M_o on $\mathcal{M}$, define*

$$\begin{aligned} M(\alpha) &= M_o + \alpha d \\ c(\beta) &= c_o + \beta h \\ \alpha^* &= \operatorname{argmin}_{\alpha\geq 0} Q(M(\alpha)) \\ \beta^* &= \operatorname{argmin}_{\beta\geq 0} J(c(\beta)) \\ M_{\beta^*} &= (A(c(\beta^*)), b(c(\beta^*)), c(\beta^*)) \end{aligned}$$

Then $Q(M_{\beta^*}) = J(c(\beta^*))$, and $Q(M_{\beta^*}) \leq Q(P(M(\alpha^*))) \leq Q(M(\alpha^*)) \leq Q(M_o)$.

3. *For M_o on $\mathcal{M}$, if $d = (F, g, h)$ is a descent direction for Q at M_o, then h is a descent direction for J at c_o.*

Proof:

1. Consider minimizing $Q(M)$ with M constrained so that $c = c_o$. The normal equations for this problem are linear and have the unique solution $(A(c_o), b(c_o))$. Hence on the constraint set $c = c_o$, $Q(M)$ has a unique global minimum at the point $(A(c_o), b(c_o), c_o) = P(M_o)$. Since M_o lies in this set, $Q(P(M_o)) \leq Q(M_o)$.

2. For $\beta \geq 0$, $M_\beta = (A(c(\beta)), b(c(\beta)), c(\beta))$ is a curve on $\mathcal{M}$ passing through M_o ($\beta = 0$) and $P(M(\alpha^*))$ ($\beta = \alpha^*$). Along this curve $Q(M_\beta) = J(c(\beta))$. Hence the minimum of Q along the curve occurs at $\beta = \beta^*$. Thus $J(c(\beta^*)) = Q(M_{\beta^*}) \leq Q(P(M(\alpha^*)))$. The other inequalities follow from part (1) and the definition of α^*.

3. Since (F, g, h) is a descent direction for Q at M_o, there exists $\alpha_o > 0$ such that $Q(M_o + \alpha d) \leq Q(M_o)$ for all $\alpha \in [0, \alpha_o]$. For $\alpha \geq 0$ let $M_\alpha = (A(c_o + \alpha h), b(c_o + \alpha h), c_o + \alpha h)$. Then for all $\alpha \in [0, \alpha_o]$, $J(c_o + \alpha h) = Q(M_\alpha) \leq Q(M_o + \alpha d) \leq Q(M_o) = J(c_o)$. The first inequality follows from part (1); the second follows from the fact that d is a descent direction for Q at M_o. ∎

Theorem 5 indicates that each step of an iterative minimization of $Q(M)$ can be improved by exploiting the formulas $A(c)$ and $b(c)$ to project the next approximation onto the manifold $\mathcal{M}$. Moreover, part (2) indicates that minimizing $J(c)$ in the direction h_k from c_k yields a greater decrease in the least-squares objective than either minimizing $Q(M)$ in the direction d_k from M_k and then projecting, or simply minimizing $Q(M)$ in the direction d_k from M_k. Other issues aside, this suggests that obtaining the least-squares estimate by iteratively minimizing $J(c)$ is potentially more efficient than a similar scheme applied to $Q(M)$. The third part of the theorem shows that at any point on the manifold $\mathcal{M}$, every descent direction for Q yields a corresponding descent direction for J. If we combine this with part (2) we see that minimization of J along this direction

will yield a smaller value of the least-squares objective function than minimizing Q in the given descent direction. Note that parts (2) and (3) of the theorem do not generally hold for M_o off the manifold $\mathcal{M}$.

Of course, J is a more complex function than Q and hence it is conceivable that the necessary computations in minimizing J are also more complex. However, as far as the gradient is concerned this is not the case. To see this, let $M(c) = (A(c), b(c), c)$. Then for each $h \in \mathbb{R}^2$,

$$DJ(c)h = D_A Q(M(c)) \cdot D_c A(c)h + D_b Q(M(c)) D_c b(c)h + D_c Q(M(c))h$$

Since $M(c)$ lies on $\mathcal{M}$, $D_A Q(M(c)) = D_b Q(M(c)) = 0$. Then from (10),

$$\begin{aligned} \nabla J(c) &= D_c Q(M(c)) \\ &= \sum_{j=1}^{N} \left(w_j' - \frac{A(c)w_j + b(c)}{c^T w_j + 1} \right)^T \frac{A(c)w_j + b(c)}{c^T w_j + 1} \frac{w_j}{c^T w_j + 1} \end{aligned} \tag{19}$$

The computation of $A(c)$ and $b(c)$ is equivalent to the computation of $\nabla_A Q$ and $\nabla_b Q$, and can be efficiently accomplished by solving the linear system (11). The computation of ∇J given $A(c)$ and $b(c)$ is equivalent to the computation of $\nabla_c Q$. Thus the computation of the gradient of J is no more complex than computing the gradient of Q.

7 Second-Derivative Methods

It is well known that minimization methods based on the second derivative of the objective function have superior rates of convergence. These methods are based on various modifications of the Newton-Raphson and Gauss-Newton schemes (see [8]). Applied to Q, these operate by setting $M_{k+1} = M_k - H(M_k)^{-1} \nabla Q(M_k)$, where $H(M_k)$ is either the Hessian of Q at M_k or a suitable approximation.

If we define $\hat{w}_j = \frac{A(c)w_j + b(c)}{c^T w_j + 1}$, we can write $Q(M) = \frac{1}{2}\sum_{j=1}^{N}(w'_j - \hat{w}_j)^T(w'_j - \hat{w}_j)$. Then

$$\begin{aligned} DQ(M) &= -\sum_{j=1}^{N}\left(w'_j - \hat{w}_j\right)^T D\hat{w}_j \\ D^2Q(M) &= \sum_{j=1}^{N} D\hat{w}_j^T D\hat{w}_j - \sum_{j=1}^{N}\left(w'_j - \hat{w}_j\right)^T D^2\hat{w}_j \end{aligned}$$

$D^2Q(M)$ is the Hessian of Q at M and the first term is the Gauss-Newton approximation of the Hessian.

It is straightforward to derive expressions for the Hessians of Q and J and their Gauss-Newton approximations. The Hessian for J is quite cumbersome since J depends on c both directly and through the dependence of $A(c)$ and $b(c)$ on c. The result is:

$$H = \sum_{j=1}^{N}\frac{1}{q_j^2(c)}\left[(\hat{w}_j - 2\varepsilon_j)^T\hat{w}_j w_j w_j^T - N_j^T(\hat{w}_j - \varepsilon_j)w_j^T\right] \tag{20}$$

where $\varepsilon_j = \left(w'_j - \hat{w}_j\right)$ and $N_j = \begin{bmatrix}\frac{\partial A}{\partial c_1}w_j + \frac{\partial b}{\partial c_1} & \frac{\partial A}{\partial c_2}w_j + \frac{\partial b}{\partial c_2}\end{bmatrix}$. The Gauss-Newton approximation to the Hessian is:

$$H_{GN} = \sum_{j=1}^{N}\frac{1}{q_j^2(c)}(N_j - \hat{w}_j w_j^T)^T(N_j - \hat{w}_j w_j^T)$$

The details of these derivations, as well as an explanation of how to compute the partial derivatives of A and b with respect to c, are lengthy and can be found in [13].

The complexity of these expressions raises the issue of obtaining efficiently computable approximations to the Hessian of J. For example, one natural approximation is to assume that A and b are independent of c so that each N_j becomes 0. This results in the approximation

$$\hat{H} = \sum_{j=1}^{N}\frac{1}{q_j^2(c)}\left[(\hat{w}_j - 2\varepsilon_j)^T\hat{w}_j w_j w_j^T\right] \tag{21}$$

In fact, this matrix is the same as the 2×2 block of partials $\frac{\partial^2 Q}{\partial c^2}$. We will see how algorithms based on this approximation and the Gauss-Newton approximation fare in the presence of different types of noise in the next section.

8 Experimental Results

We will compare various second-derivative methods for minimizing J and Q experimentally using data obtained from several pairs of natural images. In general, we will use the following framework for our second-derivative methods to minimize $J(c)$. The only difference is the approximation to the Hessian used in step 3.

Newton scheme for minimizing J.

1. Initialize $c = 0$.
2. Compute the gradient of J exactly using (19).
3. Approximate the Hessian $\frac{\partial^2 J}{\partial c^2}$ by some positive semidefinite matrix $\hat{H}$.
4. Use these quantities to update the value of c using an approximate Newton-Raphson step.
5. Use the new value of c to update the values of A and b using (14).
6. Test for convergence. Exit or return to step 2.

The initialization of $c = 0$ in step 1 is justified in practice, since the values of c for projective transformations arising from real image processing problems often have $c = O(10^{-4})$ (see Table 1). Algorithms to find a value of α in step 4 that brings about a sufficient decrease in the cost function are generally based on a backtracking and cubic interpolation strategy [8].

We implemented five minimization algorithms:

Example Number	First Image	Second Image	Number of w_j	$\hat{a}_{11}$ $\hat{a}_{21}$	$\hat{a}_{12}$ $\hat{a}_{22}$	$\hat{b}_1$ $\hat{b}_2$	$\hat{c}_1$ $\hat{c}_2$
1	Firestone1	Firestone2	70	1.1781 0.1316	-0.0640 1.1045	-173.88 0.01	0.0006 -0.0001
2	Firestone2	Firestone1	69	0.8486 -0.0976	0.0636 0.9728	146.44 -17.47	-0.0005 0.0001
3	B320fr0	B320fr1	90	0.8532 -0.0166	-0.0223 0.9639	8.51 -1.68	-0.0004 0.0002
4	Track1	Track2	30	0.9703 -0.0404	-1.5266 0.9630	83.10 -5.85	-0.0004 -0.0007
5	Atrium1	Atrium2	35	1.1146 -0.0790	0.6413 0.6171	-95.36 2.50	0.0005 -0.0008
6	Atrium2	Atrium3	33	0.7564 0.0010	-0.6599 0.8996	160.15 13.44	-0.0004 0.0009

Table 1: Information and nominal parameters for the 6 data sets.

1. GNQ: Standard Gauss-Newton applied to Q.

2. GNJ: Standard Gauss-Newton applied to J.

3. $\hat{N}$: Approximate Newton applied to J, using $\hat{H}$ from (21).

4. $QdirJ$: Approximate Newton applied to J, using the projections of search directions from Q onto the manifold, as suggested by Theorem 5.

5. NJ: Full Newton applied to J, using the actual Hessian (20).

The algorithms were compared on six pairs of natural images and associated point correspondences. Three of the image pairs were created by a rotating camera; point correspondences for these images were obtained automatically using the feature detection and matching algorithm described in Tan et al. [14]. The other three image pairs are different views of planar scenes; in these cases the point correspondences were obtained manually. For all six images there is very little noise in the correspondences. Information about the test images and the best least-squares projective transformation estimates for each example are given in Table 1.

The objective of our experiments is to investigate the performance and robustness of each of the five algorithms using the type of noisy point matchings characteristic of real image processing

applications. To accomplish this goal we modified the virtually noiseless measurements discussed above by adding noise of two different types to each correspondence. The first type was Gaussian noise of increasing variance. That is, random noise was added to each nominal correspondence $(x, y) \mapsto (x', y')$ to obtain $(\tilde{x}, \tilde{y}) \mapsto (\tilde{x}', \tilde{y}')$, where $\tilde{x} = x+n_1$, $\tilde{y} = y+n_2$, $\tilde{x}' = x'+n_3$, $\tilde{y}' = y'+n_4$ and (n_1, n_2, n_3, n_4) are independent zero-mean Gaussian random variables with variance σ^2. The second type of noise was the same, except (n_1, n_2, n_3, n_4) were drawn from a zero-mean Gaussian distribution of variance 5 with probability $1 - p$, and from a uniform distribution over $[-50, 50]$ with probability p.

The first type of noise simulates increasingly inaccurate feature correspondences. Inaccuracies in real applications could come from poor sensors, suboptimal correspondence algorithms, or coarsely subsampled data. For example, if the images were subsampled by a factor of 16 in each direction before estimating correspondence, we could expect errors in the range ± 8 pixels in the original coordinates. The second type of noise simulates a generally good correspondence algorithm with increasing probability of obtaining a non-Gaussian outlier. Such outliers can occur, for example, when a block-matching algorithm "finds" a matching block with a lower mean-squared-error than the correct block induced by camera and object motion.

Before applying the minimization algorithms to each modified data set, we normalized each of the sets $\{w_j, j = 1, \ldots, N\}$ and $\{w'_j, j = 1, \ldots, N\}$ so that the measurements are zero mean and with range approximately $[-1, 1]$. This avoids numerical instabilities introduced by data measurements that can vary by orders of magnitude. Recovering the projective transformation parameters in the original, unnormalized coordinates after the estimation is complete is accomplished by an easily derivable transformation. Details on the data normalization can be found in the appendix.

For each modified data set, the five different minimization algorithms all converged to the same projective transformation estimate. The 2-dimensional methods were initialized with $c = 0$. The 8-dimensional methods were initialized with $A = I, b = 0, c = 0$. In each case we ensured that the algorithms employed the same computational procedures and tests for conver-

gence in the appropriately-dimensioned space. Namely, the algorithm terminates when either the relative change in the gradient is small enough, i.e. $\max_{1\leq i\leq d}\left|\frac{\nabla f(x)_i \max\{|x_i|,t_i\}}{|f(x)|}\right| \leq 10^{-6}$, or the relative change in the parameters is small enough, i.e. $\frac{|\Delta x|}{\max\{|x_i|,t_i\}} \leq 10^{-6}$. Here $x = (a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2, c_1, c_2)$, $d = 8$ in the eight-dimensional case and $x = (c_1, c_2)$, $d = 2$ in the two-dimensional case, and f is the appropriately-dimensioned least-squares functional Q or J. Additionally, t_i is a "typical" value of parameter i to avoid problems with defining relative change when the parameters are small. In our tests we used $t = (1, 1, 1, 1, 100, 10, .0001, .0001)$. This choice is justified given the underlying parameters for our data set (see Table 1).

The number of floating point operations required for the three algorithms to converge with the purely Gaussian noise model is illustrated in Figure 5. Figure 5a pertains to the images taken by rotating cameras, and Figure 5b to the images of planar scenes. The x axis in each figure is the variance σ^2 of the noise added to the correspondences. The y axis in each figure represents the computational effort (i.e. number of floating point operations) required to converge to a solution, relative to Gauss-Newton on Q (the "standard" method). Each data point is the mean of 100 trials at the same noise variance with different realizations of the random variables, averaged over 3 different data sets of the same type.

We can see that using the Q search directions on J is uniformly better than doing standard Gauss-Newton on Q, and that Gauss-Newton on J is uniformly better than both. The full Newton method on J does better than the Gauss-newton method on J at higher noise variances, though worse at lower noise variances. This is consistent with the observations in Dennis [8, p. 226].

Interestingly, the approximate Newton method using the Hessian approximation in (21) is only superior to other methods at higher variances. This would indicate that while $\hat{H}$ can be computed efficiently, it is a poor approximation to the true Hessian H at low noise variances, i.e. the partial derivatives of A and b with respect to c are significant. This is confirmed by plotting the indicator $\|\hat{H} - H\|/\|H\|$ as a function of the noise variance for the first data set, illustrated in Figure 6. For comparison, we also show the indicator $\|H_{GN} - H\|/\|H\|$ for the Gauss-Newton method. We

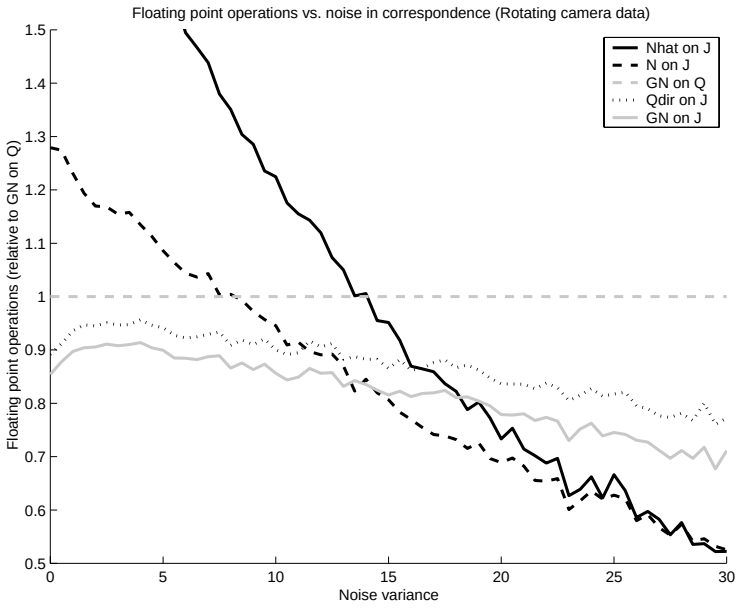
can see that in the presence of no noise, roughly 85% of the Hessian is "unapproximated" by $\hat{H}$, compared to only 0.1% in the Gauss-Newton case. Though the $\hat{N}$ iteration requires fewer floating point operations, 18 iterations were required compared to only 3 for Gauss-Newton. However, the Gauss-Newton approximation contains none of the terms in the full Hessian involving ε_j, the errors in the fitted data. Hence, as the noise variance increases, H_{GN} becomes an increasingly poor approximation. On the other hand, $\hat{H}$ contains one of the ε_j terms from the full Hessian and incrementally improves with increasing ε_j. Of course, the substantial partial derivative terms that make up most of the Hessian are still ignored.

The number of floating point operations required for the three algorithms to converge with the outlier noise model is illustrated in Figure 7. Figure 7a pertains to the images taken by rotating cameras, and Figure 7b to the images of planar scenes. The x axis in each figure is the probability p that a coordinate is an outlier. The y axis has the same interpretation as in Figure 5. Each data point is the mean of 100 trials at the same outlier probability with different realizations of the random variables, averaged over 3 different data sets of the same type.

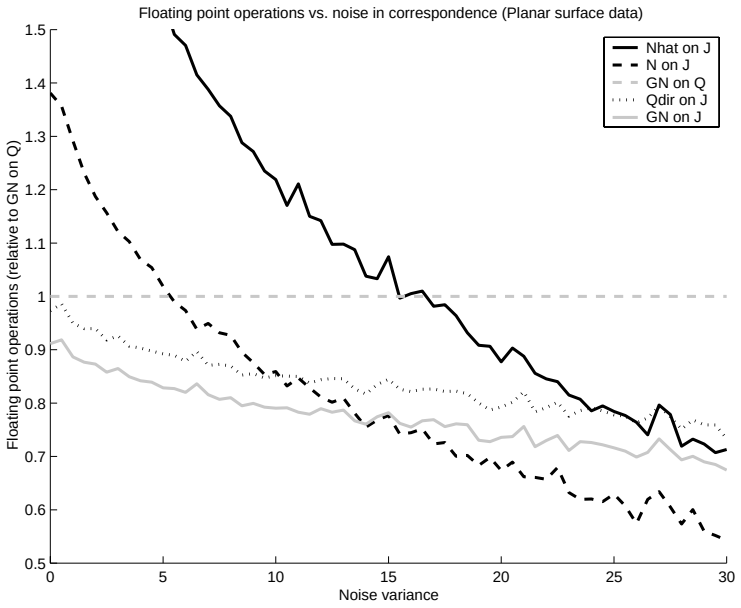
The results here again indicate the superiority of the two-dimensional algorithms. The main difference is the lower rate of decrease of the $\hat{N}$ curves, which indicates that the Gauss-Newton method on J is a better choice overall when the correspondence contains outliers. Of course, a good estimation scheme will iteratively reject outliers [15] until the noise can be well-modeled by a Gaussian distribution, and re-estimate.

9 Conclusions

The experimental results indicate that obtaining the least-squares estimate of the parameters of a projective transformation using the algorithms proposed in Section 7 to minimize $J(c)$ offers a distinct efficiency advantage over using a standard algorithm such as Gauss-Newton to minimize $Q(M)$.



(a)



(b)

Figure 5: Relative floating point operation counts for purely Gaussian noise. (a) Rotating camera data sets, (b) Planar surface data sets.

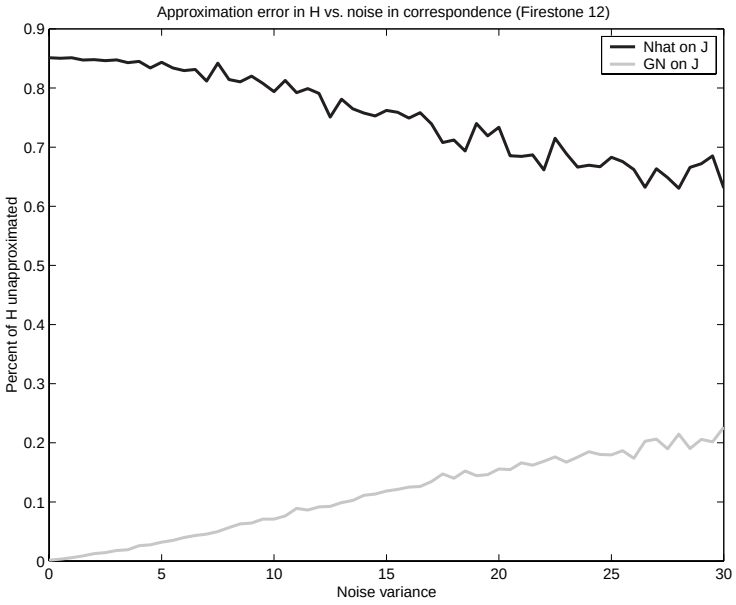
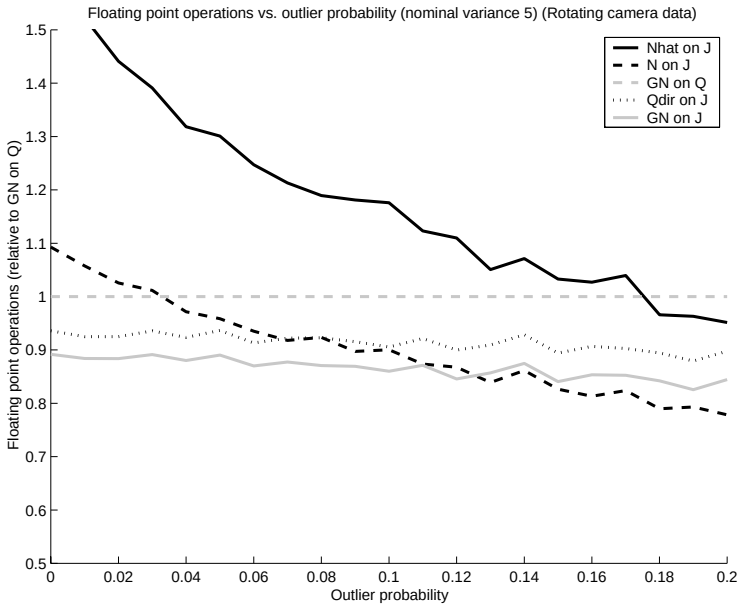


Figure 6: $\|\hat{H} - H\|/\|H\|$ as a function of noise variance for $\hat{N}$ and GNJ methods.

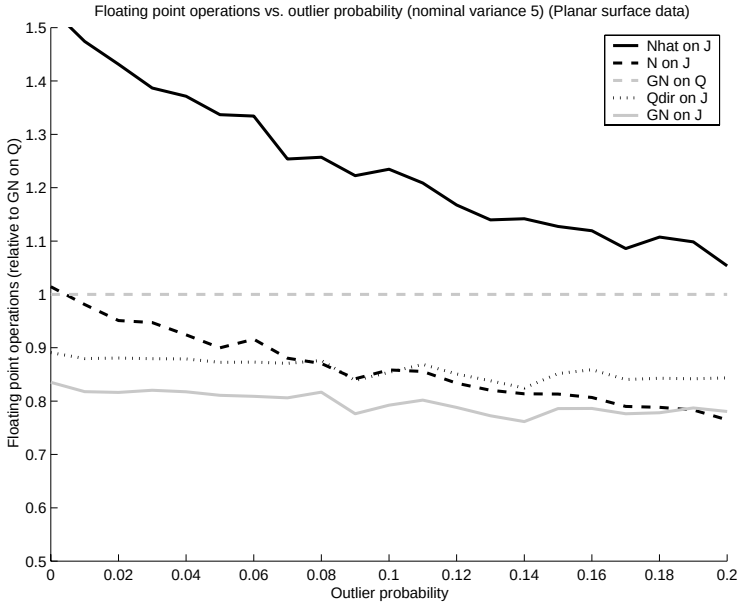
Even when it is desired to work with $Q(M)$ directly, computing $A(0)$ and $b(0)$ by (14) can provide immediate and accurate initial estimates of the parameters A and b for a minimization algorithm, removing the need for additional pre-processing to estimate the scale, rotation, and translation difference between an image pair.

Possible future research in this area includes a deeper investigation of how the relationship between the positions of the data points, the noise in their measurement, and the underlying projective transformation parameters affect the convergence of the algorithm. For example, our simulations indicate that the $\hat{N}$ algorithm presented is quite robust to high-variance noise. In fact, in terms of total floating point operations (not displayed in the figures above), its computational cost seems to decrease with noise variance while the costs of the other algorithms increase. However, we lack a rigorous analysis of why this is so.

Additionally, it may be possible to use the two-dimensional cost function $J(c)$ to analyze the existence and behavior of local minima. We have been able to construct data sets that induce a cost function $J(c)$ with multiple local minima over the region C_o, and have experimentally obtained



(a)



(b)

Figure 7: Relative floating point operation counts for Gaussian noise with outliers. (a) Rotating camera data sets, (b) Planar surface data sets.

bifurcation diagrams for the minima as the configuration of the data points is continuously varied. However, in our experience with projective transformations arising from real data, we have never observed multiple local minima of the least-squares cost functional within C_o. (It is clear from Figure 4c that many local minima exist outside of C_o).

We only address the estimation of a single projective transformation here, but there are natural extensions to the joint estimation of the projective transformations relating several images, e.g. frames of a video sequence. The composition of multiple pairwise estimates is suboptimal for the joint problem, and can lead to unstable error growth [16]. Additional issues arise when the images are constrained to form a seamless 360° panorama, as in Szeliski [4].

References

[1] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Efficient Representations of Video Sequences and Their Applications. *Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, Analysis, and Application*, Vol. 8, No. 4, pp. 327–351, May 1996.

[2] S. Mann and R.W. Picard. Video Orbits of the Projective Group: A Simple Approach to Featureless Estimation of Parameters. *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1281-1295, 1997.

[3] H.S. Sawhney, A. Ayer and M. Gorkani. Model Based 2D and 3D Dominant Motion Estimation for Mosaicing and Video Representation. *Proc. ICCV 1995*, pp. 583–590, 1995.

[4] R. Szeliski and H. Shum. Creating Full View Panoramic Image Mosaics and Environment Maps. *Computer Graphics (SIGGRAPH '97)*, pp. 251–258, August 1997.

[5] Y.P. Tan, S.R. Kulkarni and P.J. Ramadge. A New Method for Camera Motion Parameter Estimation. *Proc. ICIP 1995*, pp. 406–409, 1995.

[6] R.Y. Tsai and T.S. Huang. Estimating the Three-Dimensional Motion Parameters of a Rigid Planar Patch. *IEEE Trans. ASSP*, vol. 25, no. 6, pp. 1147–1152, December 1981.

[7] R. Radke, P. Ramadge, S. Kulkarni, T. Echigo, S. Iisaku. Recursive Propagation of Correspondences with Applications to the Creation of Virtual Video. In *Proc. ICIP 2000*, September 2000.

[8] J.E. Dennis, Jr. and R.B. Schnabel. *Numerical Methods for Unconstrianed Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[9] Y.P. Tan. *Digital Video Analysis and Manipulation.* Ph.D Thesis, Department of Electrical Engineering, Princeton University, November 1997.

[10] K. Kanatani. *Geometric Computation for Machine Vision.* Clarendon Press, 1993.

[11] R. Radke, P. Ramadge, T. Echigo, and S. Iisaku. Efficiently Estimating Projective Transformations. In *Proc. ICIP 2000*, September 2000.

[12] G.H. Golub and V. Pereyra. The Differentiation of Pseudo-Inverse and Non-Linear Least Squares Problems Whose Variables Separate. *SIAM J. Numer. Anal.*, vol. 10, pp. 413–432, 1973.

[13] R.J. Radke. *Estimation Problems in Digital Video.* Ph.D Thesis, Department of Electrical Engineering, Princeton University, June 2001.

[14] Y.P. Tan, S. Kulkarni, and P. Ramadge. Extracting Good Features for Motion Estimation. *Proc. ICIP 1996*, vol. 1, pp. 117–120, 1996.

[15] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection.* John Wiley and Sons, 1987.

[16] V. Zagorodnov and P. Ramadge. Error Stabilization in Successive Estimation of Registration Parameters. In *Proc. ICIP 2000*, Vancouver, Canada, September 2000.

Appendix

A.1 Proof of Theorem 2

Define $c(\alpha) = c^* + \alpha h$, where h is an approach vector in $\mathbb{R}^2$. Then $q_1(c(\alpha)) = \alpha h^T w_1 \neq 0$ when $\alpha \neq 0$ and $h^T w_1 \neq 0$ (i.e. the approach direction is not along the singular line). We henceforth assume that $h^T w_1$ is normalized to 1, so that $q_1(c(\alpha)) = \alpha$. (Note that $h^T w_1 = 1$ is just a line parallel to the singular line. As we decrease α, we approach the singular point c^* along lines parallel to $c^{*T} w_1 + 1 = 0$.) By solving (14) with $c(\alpha)$, we naturally define $A(\alpha)$ and $b(\alpha)$.

1. We split up the expressions (12)-(13) by separating out the first point (and noting that $q_1 = \alpha$):

$$\begin{aligned} W(c^* + \alpha h) &= \begin{bmatrix} \frac{w_1 w_1^T}{\alpha^2} + \sum_{j=2}^N \frac{w_j w_j^T}{q_j^2(\alpha)} & \frac{w_1}{\alpha^2} + \sum_{j=2}^N \frac{w_j}{q_j^2(\alpha)} \\ \frac{w_1^T}{\alpha^2} + \sum_{j=2}^N \frac{w_j^T}{q_j^2(\alpha)} & \frac{1}{\alpha^2} + \sum_{j=2}^N \frac{1}{q_j^2(\alpha)} \end{bmatrix} \\ V(c^* + \alpha h) &= \begin{bmatrix} \frac{w_1' w_1^T}{\alpha} + \sum_{j=2}^N \frac{w_j' w_j^T}{q_j(\alpha)} & \frac{w_1'}{\alpha} + \sum_{j=2}^N \frac{w_j'}{q_j(\alpha)} \end{bmatrix} \end{aligned}$$

   Here $q_j(\alpha) = (c^* + \alpha h)^T w_j + 1$. We now rewrite the defining equation (11) as:

$$\begin{bmatrix} A(\alpha) & b(\alpha) \end{bmatrix} \left[\frac{1}{\alpha^2} \begin{bmatrix} w_1 w_1^T & w_1 \\ w_1^T & 1 \end{bmatrix} + W_2(\alpha) \right] = \left[\frac{1}{\alpha} w_1' p^T + V_2(\alpha) \right] \tag{22}$$

   where

$$W_2(\alpha) = \begin{bmatrix} \sum_{j=2}^N \frac{w_j w_j^T}{q_j^2(\alpha)} & \sum_{j=2}^N \frac{w_j}{q_j^2(\alpha)} \\ \sum_{j=2}^N \frac{w_j^T}{q_j^2(\alpha)} & \sum_{j=2}^N \frac{1}{q_j^2(\alpha)} \end{bmatrix}$$

$$V_2(\alpha) \;=\; \left[\begin{array}{cc} \sum_{j=2}^{N} \frac{w_j' w_j^T}{q_j(\alpha)} & \sum_{j=2}^{N} \frac{w_j'}{q_j(\alpha)} \end{array}\right]$$

We note that as $\alpha \to 0$, $W_2(\alpha)$ and $V_2(\alpha)$ converge to well-defined finite matrices $W_2(0)$ and $V_2(0)$. In the following, we will use the notations W_2, V_2 with the understanding that they are functions of α. While it is true that for $\alpha > 0$, $W_2(\alpha)$ and $V_2(\alpha)$ are also functions of h, the limiting values $W_2(0)$ and $V_2(0)$ are independent of h.

Taking (22) and isolating $A(\alpha)$ and $b(\alpha)$ on the left-hand side, we have

$$\left[\begin{array}{cc} A(\alpha) & b(\alpha) \end{array}\right] = \left[\frac{1}{\alpha} w_1' p^T + V_2\right] \left[\frac{1}{\alpha^2} \left[\begin{array}{cc} w_1 w_1^T & w_1 \\ w_1^T & 1 \end{array}\right] + W_2\right]^{-1} \tag{23}$$

Here we have introduced the abbreviation $p = [w_1^T 1]^T$. First consider the matrix that is inverted in (23) above. Using the matrix inversion lemma, we can write

$$\begin{aligned} \left[\frac{1}{\alpha^2} \left[\begin{array}{cc} w_1 w_1^T & w_1 \\ w_1^T & 1 \end{array}\right] + W_2\right]^{-1} &= \left[\frac{1}{\alpha} p \frac{1}{\alpha} p^T + W_2\right]^{-1} \\ &= W_2^{-1} - \frac{1}{\alpha^2} W_2^{-1} p \left[1 + \frac{1}{\alpha^2} p^T W_2^{-1} p\right]^{-1} p^T W_2^{-1} \\ &= W_2^{-1} - W_2^{-1} p \left[\alpha^2 + p^T W_2^{-1} p\right]^{-1} p^T W_2^{-1} \\ &= W_2^{-1} - \frac{W_2^{-1} p p^T W_2^{-1}}{\alpha^2 + p^T W_2^{-1} p} \\ &= W_2^{-1} \left[I - \frac{p p^T W_2^{-1}}{\alpha^2 + p^T W_2^{-1} p}\right] \end{aligned}$$

Therefore, we can rewrite (23) as

$$\begin{aligned} \left[\begin{array}{cc} A(\alpha) & b(\alpha) \end{array}\right] &= \left[\frac{1}{\alpha} w_1' p^T + V_2\right] W_2^{-1} \left[I - \frac{p p^T W_2^{-1}}{\alpha^2 + p^T W_2^{-1} p}\right] \\ &= w_1' p^T W_2^{-1} \left[\frac{\frac{1}{\alpha}\left(\alpha^2 + p^T W_2^{-1} p\right) I - \frac{1}{\alpha} p p^T W_2^{-1}}{\alpha^2 + p^T W_2^{-1} p}\right] \\ &\quad + V_2 W_2^{-1} \left[I - \frac{p p^T W_2^{-1}}{\alpha^2 + p^T W_2^{-1} p}\right] \\ &= w_1' p^T W_2^{-1} \frac{\alpha}{\alpha^2 + p^T W_2^{-1} p} + V_2 W_2^{-1} \left[I - \frac{p p^T W_2^{-1}}{\alpha^2 + p^T W_2^{-1} p}\right] \end{aligned}$$

Letting $\alpha \to 0$ in the above, we obtain

$$\begin{aligned} \left[\begin{array}{cc} A_o & b_o \end{array}\right] &= \lim_{\alpha \to 0} \left[\begin{array}{cc} A(\alpha) & b(\alpha) \end{array}\right] \\ &= V_2 W_2^{-1} \left[I - \frac{p p^T W_2^{-1}}{p^T W_2^{-1} p}\right] \end{aligned} \tag{24}$$

2. Consider the minimization problem

$$\begin{array}{ll} \min\limits_{A,b} & \frac{1}{2}\sum\limits_{j=2}^{N}\left(w_j' - \frac{Aw_j+b}{c^{*T}w_j+1}\right)^T\left(w_j' - \frac{Aw_j+b}{c^{*T}w_j+1}\right) \\ s.t. & Aw_1 + b = 0 \end{array} \tag{25}$$

The normal equations for the constrained problem are:

$$\begin{aligned} A\sum_{j=2}^{N}\frac{w_j w_j^T}{(c^{*T}w_j+1)^2} + b\sum_{j=2}^{N}\frac{w_j^T}{(c^{*T}w_j+1)^2} - \sum_{j=2}^{N}\frac{w_j' w_j^T}{c^{*T}w_j+1} + \lambda w_1^T &= 0 \\ A\sum_{j=2}^{N}\frac{w_j}{(c^{*T}w_j+1)^2} + b\sum_{j=2}^{N}\frac{1}{(c^{*T}w_j+1)^2} - \sum_{j=2}^{N}\frac{w_j'}{c^{*T}w_j+1} + \lambda &= 0 \\ Aw_1 + b &= 0 \end{aligned}$$

Here λ is a Lagrange multiplier in $\mathbb{R}^2$. Rewriting these normal equations in the notation of the previous section, we obtain

$$\left[\begin{array}{cc} A & b \end{array}\right] = V_2 W_2^{-1} - \lambda p^T W_2^{-1} \tag{26}$$

$$\left[\begin{array}{cc} A & b \end{array}\right] p = 0 \tag{27}$$

It is easy to see that (26) is satisfied by the choice of (A_o, b_o) in (24), with

$$\lambda = \frac{V_2 W_2^{-1} p}{p^T W_2^{-1} p}$$

Furthermore, (A_o, b_o) satisfy the constraint equation (27). Hence, by uniqueness of the solution of the linear least-squares problem, we conclude that along singular lines, the solution of (11) converges to the solution of the constrained minimization problem (25) posed over the data set minus the offending point.

3. From the expression (24), we can also obtain an expression for the point to which (A_o, b_o, c^*) maps w_1:

$$\begin{aligned} \lim_{\alpha\to 0}\frac{A(\alpha)w_1 + b(\alpha)}{(c^* + \alpha h)^T w_1 + 1} &= \lim_{\alpha\to 0}\frac{1}{\alpha}\left[\begin{array}{cc} A(\alpha) & b(\alpha) \end{array}\right] p \\ &= \lim_{\alpha\to 0}\frac{1}{\alpha} w_1' p^T W_2^{-1}\frac{\alpha}{\alpha^2 + p^T W_2^{-1} p} \\ &\quad + V_2 W_2^{-1}\left[I - \frac{p p^T W_2^{-1}}{\alpha^2 + p^T W_2^{-1} p}\right] p \\ &= \lim_{\alpha\to 0} w_1' p^T W_2^{-1}\frac{p}{\alpha^2 + p^T W_2^{-1} p} \end{aligned}$$

$$
\begin{aligned}
& + V_2 W_2^{-1} \left[\frac{1}{\alpha} p - \frac{1}{\alpha} \frac{p p^T W_2^{-1} p}{\alpha^2 + p^T W_2^{-1} p} \right] \\
= & \lim_{\alpha \to 0} w_1' \frac{p^T W_2^{-1} p}{\alpha^2 + p^T W_2^{-1} p} + V_2 W_2^{-1} \frac{\alpha p}{\alpha^2 + p^T W_2^{-1} p} \\
= & w_1'
\end{aligned}
$$

A.2 Data Normalization

To avoid numerical instabilities introduced by data measurements that vary by orders of magnitude, it is generally wise to normalize the data before processing it. Hence, we need to understand how the solution to the least-squares problem using the normalized data is related to the solution of the problem in the original coordinates. To this end, we present the following lemma, which is easily proven.

Lemma 1 *Consider the data sets given by*

$$
\begin{aligned}
z_j &= Q w_j + t \\
z_j' &= R w_j' + t'
\end{aligned}
$$

for $t, t' \in \mathbb{R}^2$, $Q, R \in GL(2)$, and $j = 1, \ldots, N$. If $\hat{M} = (\hat{A}, \hat{b}, \hat{c})$ is the minimizer of

$$
Q(M) = \frac{1}{2} \sum_{j=1}^{N} \left(w_j' - \frac{A w_j + b}{c^T w_j + 1} \right)^T \left(w_j' - \frac{A w_j + b}{c^T w_j + 1} \right)
$$

then the minimizer $\tilde{M} = (\tilde{A}, \tilde{b}, \tilde{c})$ of

$$
\tilde{Q}(M) = \frac{1}{2} \sum_{j=1}^{N} \left(z_j' - \frac{A z_j + b}{c^T z_j + 1} \right)^T \left(z_j' - \frac{A z_j + b}{c^T z_j + 1} \right)
$$

is given by

$$
(\tilde{A}, \tilde{b}, \tilde{c}) = \left(\frac{R \hat{A} Q^{-1} + t' \hat{c}^T Q^{-1}}{1 - \hat{c}^T Q^{-1} t}, \frac{R \hat{b} - R \hat{A} Q^{-1} t}{1 - \hat{c}^T Q^{-1} t}, \frac{Q^{-T} c}{1 - \hat{c}^T Q^{-1} t} \right)
$$

In other words,

$$
(\hat{A}, \hat{b}, \hat{c}) = \left(\frac{R^{-1}(\tilde{A} - t' \tilde{c}^T) Q}{1 + \tilde{c}^T t}, \frac{R^{-1}(\tilde{b} + \tilde{A} t)}{1 + \tilde{c}^T t} - R^{-1} t', \frac{Q^T \tilde{c}}{1 + \tilde{c}^T t} \right) \tag{28}
$$

In practice, we normalize the data so that the measurements are zero mean with range approximately $[-1, 1]$. This corresponds to a choice of

$$
\begin{aligned}
t &= -\tfrac{\mu}{\alpha} \quad t' = -\tfrac{\mu'}{\alpha'} \\
Q &= \tfrac{1}{\alpha} I \quad R = \tfrac{1}{\alpha'} I
\end{aligned}
$$

where

$$\begin{array}{ll} \mu = \frac{1}{N}\sum_{j=1}^{N} w_j & \mu' = \frac{1}{N}\sum_{j=1}^{N} w'_j \\ \alpha = \frac{1}{2}\left(\max_j |x_j - \mu_x| + \max_j |y_j - \mu_y|\right) & \alpha' = \frac{1}{2}\left(\max_j |x'_j - \mu'_x| + \max_j |y'_j - \mu'_y|\right) \end{array}$$

For this choice of (t, t', Q, R), we can rewrite (28) as

$$\hat{A} = \frac{\frac{1}{\alpha}\left(\alpha'\tilde{A} + \mu'\tilde{c}\right)}{1 - \frac{\mu}{\alpha}\tilde{c}} \qquad \hat{b} = \frac{\alpha'(\tilde{b} - \tilde{A}\frac{\mu}{\alpha})}{1 - \frac{\mu}{\alpha}\tilde{c}} + \mu' \qquad \hat{c} = \frac{\tilde{c}}{\alpha - \mu\tilde{c}}$$