EFFICIENTLY ESTIMATING PROJECTIVE TRANSFORMATIONS

Richard Radke, Peter Ramadge

Department of Electrical Engineering
Princeton University
Princeton, NJ 08544

Tomio Echigo

IBM Tokyo Research Laboratory
1623-14 Shimotsuruma
Yamato-shi, Kanagawa, Japan

Shun-ichi Iisaku

Communication Research Laboratory
Ministry of Posts and Telecommunications
4-2-1 Nukuikita-machi
Koganei-shi, Tokyo, Japan

1. INTRODUCTION

The estimation of the parameters of a projective transformation that relates the coordinates of two image planes is a standard problem that arises in image and video mosaicking, virtual video, and problems in computer vision [1], [2], [3], [4], [5], [6]. This problem is often posed as a least squares minimization problem based on a finite set of noisy point samples of the underlying transformation. While in some special cases this problem can be solved using a linear approximation, in general, it results in an 8-dimensional nonquadratic minimization problem that is solved numerically using an 'off-the-shelf' procedure such as the Levenberg-Marquardt algorithm [7].

We show that the general least squares problem for estimating a projective transformation can be analytically reduced to a 2-dimensional nonquadratic minimization problem. Moreover, we provide both analytical and experimental evidence that the minimization of this function is computationally attractive. We propose a particular algorithm that is a combination of a projection and an approximate Gauss-Newton scheme, and experimentally verify that this algorithm efficiently solves the least squares problem.

2. PROBLEM STATEMENT

We wish to estimate the parameters $A \in \mathbf{R}^{2\times 2}$, $b, c \in \mathbf{R}^2$ of a projective transformation $g_M(w) = \frac{Aw+b}{cw+1}$. Here we regard b as a 2×1 matrix, and c as a 1×2 matrix, and let $M = (A, b, c)$. Our objective is to select the parameters so that g_M best matches a given set of point mappings $\{w_j \mapsto w_j' \in \mathbf{R}^2, j = 1, \ldots, N\}$. A special case arises when the data consists of noisy samples of a fixed but unknown projective transformation $g_{M^*}\colon w_j' = g_{M^*}(w_j) + e_j$, $j = 1 \ldots N$. Here $e_j \in \mathbf{R}^2$ is the error in the measurement of $g_{M^*}(w_j)$. In this case we seek an estimate M of M^*.

The singular line of the transformation $Aw + b/(cw+1)$ is the set $\{w\colon cw = -1\}$. Along this line it is assumed that $Aw + b \neq 0$, i.e., the matrix $\begin{pmatrix} A & b \\ c & 1 \end{pmatrix} \in GL(3)$. We denote this by writing $M \in GL(3)$, where M is now the triple (A, b, c) or the above matrix, depending on context.

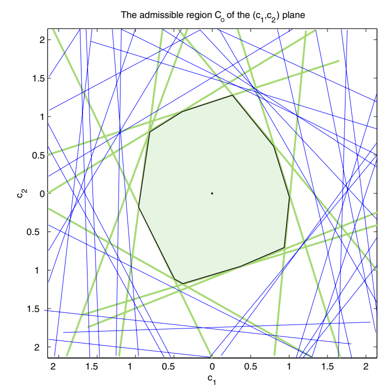

Fig. 1. The admissible region C_o of the (c_1, c_2) plane generated by data points from actual images. Thin blue lines represent singular lines; thick green lines are singular lines which actively bound the admissible region.

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 $cw + 1 > 0$ for all $w \in W$. This is equivalent to the requirement that $cw_j + 1 > 0$, $j = 1, \ldots, N$. This defines an open convex set $C_o \subset \mathbf{R}^2$ 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 $\mathcal{A} = \{(A, b, c)\colon A \in \mathbf{R}^{2\times 2}, b \in \mathbf{R}^2, c \in C_o\}$. Note that admissibility does not require $M \in GL(3)$. Figure 1 illustrates the admissible region C_o generated by data points from actual images.

3. THE LEAST SQUARES ESTIMATE

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

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

This research was partially supported by grants from the IBM Tokyo Research Laboratory and the New Jersey Center for Multimedia Research.

over admissible $M = (A, b, c)$.

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

For $c \in C_o$ define $A(c)$ and $b(c)$ as solutions to a linear system:

$$[\ A(c) \quad b(c)\ ]W(c) = V(c) \tag{2}$$

where $q_j(c) = cw_j + 1$ and

$$W(c) = \begin{bmatrix} \sum_{j=1}^{N} \frac{w_j w_j^T}{q_j^2(c)} & \sum_{j=1}^{N} \frac{w_j}{q_j^2(c)} \\ \sum_{j=1}^{N} \frac{w_j^T}{q_j^2(c)} & \sum_{j=1}^{N} \frac{1}{q_j^2(c)} \end{bmatrix} \tag{3}$$

$$V(c) = \begin{bmatrix} \sum_{j=1}^{N} \frac{w'_j w_j^T}{q_j(c)} & \sum_{j=1}^{N} \frac{w'_j}{q_j(c)} \end{bmatrix} \tag{4}$$

We assume that the points $\{w_j : j = 1, \ldots, N\}$ are not colinear in $\mathbf{R}^2$. This ensures that $W(c)$ is positive definite and hence that $A(c)$ and $b(c)$ are defined for all $c \in C_o$.

We can now state our first result.

Theorem 1 *Assuming that the points w_j, $j = 1, \ldots, N$ are noncolinear, 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 $\mathbf{R}^{2\times 2} \times \mathbf{R}^2 \times C_o$.*

Proof: Since $\hat{M}$ minimizes (1), 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:

$$\hat{A} \sum_j \frac{w_j w_j^T}{(\hat{c}w_j+1)^2} + \hat{b} \sum_j \frac{w_j^T}{(\hat{c}w_j+1)^2} - \sum_j \frac{w'_j w_j^T}{\hat{c}w_j+1} = 0 \tag{5}$$

$$\hat{A} \sum_j \frac{w_j}{(\hat{c}w_j+1)^2} + \hat{b} \sum_j \frac{1}{(\hat{c}w_j+1)^2} - \sum_j \frac{w'_j}{\hat{c}w_j+1} = 0 \tag{6}$$

$$\sum_j \left(w'_j - \left(\frac{\hat{A}w_j+\hat{b}}{\hat{c}w_j+1} \right) \right)^T \left(\frac{\hat{A}w_j+\hat{b}}{\hat{c}w_j+1} \right) \frac{w_j}{\hat{c}w_j+1} = 0 \tag{7}$$

Rewriting the normal equations (5) and (6) yields (2). Thus $(\hat{A}, \hat{b}, \hat{c}) = (A(\hat{c}), b(\hat{c}), \hat{c})$ and the theorem follows. ∎

In view of Theorem 1, define $J : C_o \to \mathbf{R}$ by

$$J(c) = \frac{1}{2} \sum_{j=1}^{N} \left(w'_j - \frac{A(c)w_j+b(c)}{cw_j+1} \right)^T \left(w'_j - \frac{A(c)w_j+b(c)}{cw_j+1} \right) \tag{8}$$

$J(c)$ is simply the least squares cost function restricted to the manifold $\mathcal{M}$. 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. Figure 2 illustrates the cost function J graphed over the region C_o for data points from actual images. Within C_o, the cost function has a single minimum located at the bottom of a deep bowl.

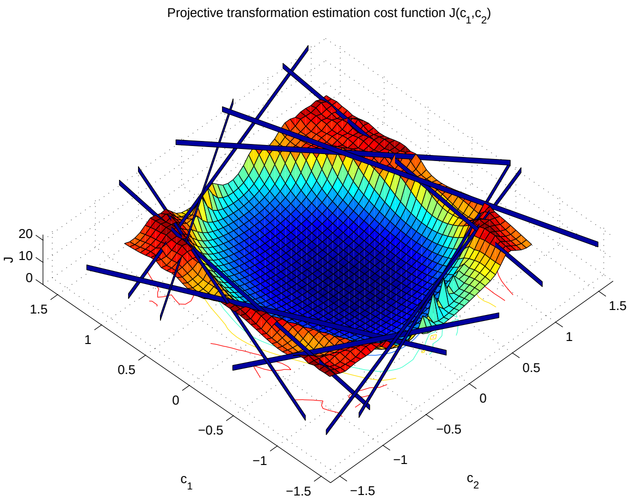

Fig. 2. The cost function $J(c_1, c_2)$ for data points from actual images. The dark blue lines are the singular lines which actively bound C_o.

For c^* on exactly one singular line, say $c^* w_1 + 1 = 0$, the matrices $W(c^*)$ and $V(c^*)$ which would define $(A(c^*), b(c^*))$ are infinite. However, we can show that the limiting solution of (2) as c approaches c^* is the solution to the well-defined constrained least-squares problem

$$\min_{A,b} \frac{1}{2} \sum_{j=2}^{N} \left(w'_j - \frac{Aw_j+b}{c^* w_j+1} \right)^T \left(w'_j - \frac{Aw_j+b}{c^* w_j+1} \right) \tag{9}$$

$$s.t.\ Aw_1 + b = 0 \tag{10}$$

The limiting value of $\frac{A(c)w_1+b(c)}{cw_1+1}$ as c approaches c^* can be shown to be w'_1. Therefore, the cost function $J(c_1, c_2)$ is finite along the singular lines, which can be seen in Figure 2. However, along singular lines the resulting least-squares projective transformation estimates are not members of $GL(3)$.

For c^* at the intersection of exactly two singular lines, the limiting solution of (2) as c approaches c^* is the solution to a constrained least-squares problem similar to (9)-(10) over the data set minus the two offending points. Intersections of three singular lines are rare and only occur when three data points are colinear.

4. OBTAINING THE LEAST-SQUARES ESTIMATE

Typical algorithms for the minimization of (1) operate iteratively 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 $Q(M_{k+1}) \leq Q(M_k)$. d_k is typically related to the gradient of the objective function evaluated at M_k.

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

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

$$\begin{aligned} M(\alpha) &= M_o + \alpha d && (11)\\ c(\beta) &= c_o + \beta h && (12)\\ \alpha^* &= \operatorname{argmin}_{\alpha \geq 0} Q(M(\alpha)) && (13)\\ \beta^* &= \operatorname{argmin}_{\beta \geq 0} J(c(\beta)) && (14)\\ M_{\beta^*} &= (A(c(\beta^*)), b(c(\beta^*)), c(\beta^*)) && (15)\end{aligned}$$

   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 $A(c_o)$ and $b(c_o)$. 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).

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 2 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 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.

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 $h \in \mathbf{R}^2$,

$$\begin{aligned} DJ(c)h &= D_A Q(M(c)) \cdot D_c A(c) h && (16)\\ &\quad + D_b Q(M(c)) D_c b(c) h + D_c Q(M(c)) h \end{aligned}$$

Since $M(c)$ lies on $\mathcal{M}$, $D_A Q(M(c)) = D_b Q(M(c)) = 0$. Then from (7),

$$\begin{aligned} \nabla J(c) &= D_c Q(M(c)) && (17)\\ &= \sum_j \left(w_j' - \tfrac{A(c)w_j + b(c)}{cw_j + 1} \right)^T \tfrac{A(c)w_j + b(c)}{cw_j + 1} \tfrac{w_j}{cw_j + 1} \end{aligned}$$

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 (2). 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.

It is well known that minimization methods based on the second derivative of the object function have superior rates of convergence. These methods are based on various modifications of the Newton-Raphson and Gauss-Newton schemes. 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. The Levenberg-Marquardt algorithm is a common modification of the basic Gauss-Newton scheme. Write

$$Q(M) = \tfrac{1}{2} \sum_j \left(w_j' - g_j(M) \right)^T \left(w_j' - g_j(M) \right) \qquad (18)$$

where $g_j(M) = (Aw_j + b)/(cw_j + 1)$. Then

$$\begin{aligned} DQ(M) &= -\sum_j \left(w_i' - g_i(M) \right)^T Dg_i(M) && (19)\\ D^2 Q(M) &= \sum_j Dg_i(M)^T Dg_i(M) && (20)\\ &\quad - \sum_j \left(w_i' - g_i(M) \right)^T D^2 g_i(M) \end{aligned}$$

$D^2Q(M)$ 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 Gauss-Newton approximation to the Hessian of Q and for the Hessian of J and its Gauss-Newton approximation. 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. Limited space precludes the inclusion of these equations here. We note, however, that the complexity of the expressions raises the issue of obtaining efficiently computable approximations to the Hessian of J. In this regard we propose the following algorithm.

Proposed algorithm for minimizing J:

Compute the gradient of J exactly using (17) and approximate the Hessian of J by assuming that A and b do not depend on c. Then use these quantities to update the value of c using an (approximate) Newton-Raphson step. Finally, use the new value of c to update the values of A and b using the formulas for $A(c)$ and $b(c)$.

Experimental results on the performance of this proposed algorithm are reported in the next section.

5. EXPERIMENTAL RESULTS

We coded three algorithms in MATLAB: standard Gauss-Newton applied to Q, standard Gauss-Newton applied to J, and the algorithm proposed above. In each case we ensured that the algorithms employed the same computational procedures and tests. The algorithms were compared on 6 sets of point correspondences, each obtained from pairs of natural images related by projective transformations. The number of point correspondences and the resultant estimated parameters for each example are shown in the table below, and the computational requirements of the algorithms for each of the 6 examples are shown in the figure below. In each example, all three algorithms converged to the same projective transformation estimate. Averaged over the 6 examples, the proposed algorithm was 2.7 times more efficient per sample point than the standard Gauss-Newton method applied to Q.

Ex. #	# 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	47	1.0855 -0.0013	0.0444 0.741	64.063 -34	0.000169 -0.000163
2	47	1.073 -0.0089	0.386 0.6583	-32.33 -30.63	0.00069 -0.00112
3	47	0.87 0.014	-0.205 1.103	78.51 4.7	-0.00058 0.00202
4	26	0.065 -0.21	1.001 0.15	-65.218 95.23	-0.00157 -0.0013
5	32	1.028 0.0337	-4.164 2.255	426.76 27.377	-0.000096 0.00817
6	18	0.111 -0.23	0.98 0.125	-89.22 111.1	-0.001586 -0.000862

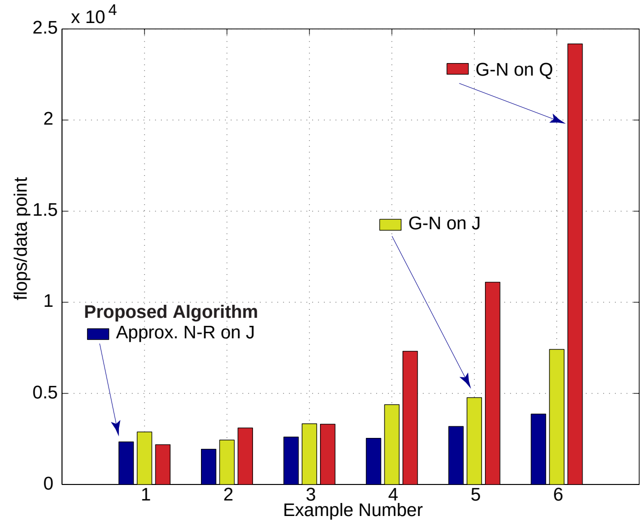

Fig. 3. Floating-point operation counts for the three methods.

6. CONCLUSIONS

Obtaining the least squares estimate of the parameters of a projective transformation using the algorithm proposed in section 4 to minimize J offers a worthwhile efficiency advantage. The experimental results indicate that, in general, minimizing J using the proposed algorithm does not incur a significant computational penalty over using a standard algorithm such as Levenberg-Marquardt to minimize Q and in certain cases it can offer a distinct efficiency advantage.

7. REFERENCES

[1] R.Y. Tsai and T.S. Huang, "Estimating the three-dimensional motion parameters of a rigid planar patch," IEEE Trans. ASSP, 25(6), Dec. 1981, 1147-1152.

[2] S. Mann and R.W. Picard, "Video orbits of the projective group: a simple approach to featureless estimation of parameters," IEEE Trans. IP, 6(9), 1281-1295.

[3] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosiacs and environment maps," Proc. SIGGRAPH, 1997.

[4] Y.-P. Tan, S. R. Kulkarni and P.J. Ramadge, "A new method for camera motion parameter estimation," Proc. IEEE ICIP, 1995, 406-409.

[5] H.S. Sawhney, A. Ayer and M. Gorkani, "Model based 2D and 3D dominant motion estimation for mosaicing and video representation," Proc. ICCV, 1995, 583-590.

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

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