Projective Tracking Based on Second-order Optimization on Lie Manifolds

Guangwei Li¹, ², ⁴, Yunpeng Liu¹, ⁴, Jian Yin³, Zelin Shi¹

1. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
   E-mail: guangweili@yahoo.com.cn
2. Management Science and Engineering Department of Qingdao University, Qingdao 266071, China
3. The Research Institute on General Development and Argumentation of Air Force, Beijing 100076, China
4. Graduate School of the Chinese Academy of Sciences, Beijing 100039, China

Abstract: Template tracking based on the space transformation model can usually be reduced to solve a nonlinear least squares optimization problem over a Lie manifold of parameters. The algorithm on the vector space has more limitations when it concerns the nonlinear projective warps. Exploiting the special structure of Lie manifolds allows one to devise a method for optimizing on Lie manifolds in a computationally efficient manner. The mapping between a Lie group and its Lie algebra can make us to utilize the specific properties of the target tracking to propose a second-order minimization tracking method. This approach needs not calculating the Hessian matrix and reduces the computation complexity. The comparative experiments with the algorithm based on the vector space and the Gauss-Newton algorithm based on the Lie algebra parameterization validate the feasibility and high effectiveness of our method.

Key Words: projective transformation; target tracking; Lie manifolds; geometric optimization

1 INTRODUCTION

The manifolds optimization algorithm[1], a novel approach to solve the constrained problem, was proposed in the 1970s-1980s. The fundamental idea of this approach is to regard the constrained sets as one underlying manifold and to exploit the geometry of the underlying parameter space. Lie Groups are important cases of continuous groups in which, in addition to the group structure, the set underlying the group has the structure of a differentiable manifold. Exploiting the deep connection between the Lie group and its Lie algebra, the geometric optimization approach based on Lie Groups theory switches the constrained nonlinear problems into equivalently unconstrained problems, thereby significantly reducing the computational complexity. The optimization algorithms based on Lie Groups and Riemannian manifolds[2] have already been applied to robot control and machine learning. Recent years have also witnessed the rich achievements in the fields of signal processing, computer vision and pattern recognition[3][4].

Target tracking is of interest to scientists and engineers of various fields: computer vision, pattern recognition, and robotics[5]. One of the remarkable problems both in theory and in technique is how to cope with the geometric warps. The geometric warps often be modeled by affine transformation, or even by projective transformation[6] when necessary. In traditional tracking approaches, there are two major groups: either the tracking is performed from local correspondences (feature-based approaches) or from template correspondences (template-based approaches). Feature-based approaches use local features such as points, line segments, edges, or regions. Such techniques are naturally less sensitive to partial occlusions as they are based on local correspondences. If several correspondences are missing, the pose is still computable. On the other hand, global or template-based approaches take the template as a whole. The strength of these methods lies in their ability to treat complex templates or patterns that cannot be modeled by local features. Within the classical space transformation tracking framework[7] proposed by Hager et al, Baker proposed the inverse-composition algorithm[8] not only to compute the Hessian matrix and the gradient matrix offline but also to improve the efficiency by improving the iterative structure. A projective tracking approach[9] based on the matrix parameterization was presented by Buenaposada et al in terms of the forward-addition algorithm on the vector space, which is called the Vector-GN algorithm in this paper. However, these algorithms can not utilize the Lie transformation group structure sufficiently and leave room to improve the performance of the tracking algorithms. Based on the one-order Taylor expansion series on Lie manifolds, Eduardo et al proposed a novel projective target tracking approach[10]. The performance, such as the tracking precision and rate, is better than that of the tracking method based on the matrix parameterization. We point that the mapping between a Lie group and its Lie algebra can make us to utilize the specific properties of the target tracking to propose a second-order minimization tracking method. This approach needs not calculating the Hessian matrix and reduces the computation complexity to improve the performance further.

The paper is organized as follows. After a brief introduction to the Lie Groups theory, a general method for optimizing a function on a Lie manifold is outlined in section 2. Section 3 investigates the efficient template tracking algorithm based on the second-order minimization on Lie manifolds. Some comparative results are shown for illustration and...
verification in section 4. Finally, section 5 concludes the investigation and proposes some further work.

2 LIE GROUPS THEORY AND FRAMEWORK FOR OPTIMIZATION ON LIE MANIFOLDS

The Lie Groups theory[11] and the manifolds optimization algorithm build the basis for our efficient projective tracking method.

2.1 Lie Groups Theory

A Lie group is a group endowed with the smooth manifold structure, and its group multiplicative operation is denoted by \( \circ \). The tangent space at the identity element \( e \) of Lie group \( M \) is denoted by \( T_e M \). The vector space \( (T_e M, [\cdot, \cdot]) \) equipped with a bilinear bracket operation is a Lie algebra denoted by \( \Lambda(M) \). There exists an open neighborhood \( W \) of \( 0 \) in Lie algebra \( \Lambda(M) \) and an open neighborhood \( U \) of \( e \) in \( M \) such that the Lie exponential map between the Lie group and Lie algebra is an analytic diffeomorphism of \( W \) onto \( U \).

The space of all \( n \times n \) nonsingular real matrices forms a Lie group, called the general linear group denoted by \( \text{GL}(n, R) \). Its algebra is usually denoted by \( gl(n, R) \), the set of all real square matrices. Being a sub-group of \( \text{GL}(n, R) \), the special linear group \( \text{SL}(n, R) \) is the space of all real \( n \times n \) matrices \( H \) satisfying \( \det H = 1 \). Its Lie algebra denoted by \( \text{sl}(n, R) \) consists of the real matrices of trace zero. What we concern in this paper is \( \text{SL}(3, R) \) whose Lie algebra is \( \text{sl}(3, R) \) with the following basis vectors:

\[
e_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},
\]

\[
e_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_5 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},
\]

\[
e_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\]

For the matrix Lie groups, the group operation is matrix multiplication. The Lie bracket operation is \([A, B] = AB - BA\) and the Lie exponential map of a matrix \( X \in gl(n, R) \) is computed by the formula

\[
\exp X = \sum_{n=1}^{\infty} \frac{X^n}{n!}.
\]

2.2 Framework for Optimization on Lie Manifolds

If a Lie manifold is embedded in Euclidean space to be a sub-manifold, the optimization problem on it can often become a classical constrained optimization. The conventional approach of dealing with the structure of the group is to use Lagrange multipliers. Based on the geometric optimization theory, we use local canonical coordinates to represent parameters and intrinsically take care of the geometric structure of Lie Groups to allow the use of unconstrained optimization routines.

Let \( x \) be a point in the neighborhood of \( t \in M \), there exists \( \omega = \sum^n_i v_i e_i \in \Lambda(M) \) such that

\[
x = t \circ \exp(\omega) = t \circ \exp(\sum^n_i v_i e_i)
\]

where \( v = (v_1, v_2, \cdots, v_n)^T \); \( e_i (i = 1, \cdots, n) \) is the basis of Lie algebra \( \Lambda(M) \).

Then, the Taylor series of a smooth function \( \phi(\cdot) \) on Lie manifold \( M \) is obtained

\[
\phi(t \circ \exp(\omega)) = \phi(t) + J^T_f v + \frac{1}{2} v^T H^o_f v + O(\|v\|^2)
\]

where \( [J^o_f]_{ij} = \frac{\partial}{\partial v_i} \phi(t \circ \exp(\omega)) |_{v=0} \);

and \( [H^o_f]_{ij} = \frac{\partial^2}{\partial v_i \partial v_j} \phi(t \circ \exp(\omega)) |_{v=0} \).

The Taylor series (3) allows us to construct various optimization algorithms on Lie groups by generalizing algorithms on vector space. For example, the classical Newton-Raphson method adopts the following intrinsic update step

\[
t \leftarrow t \circ \exp(\omega)
\]

where \( v \) solves \( H^o_f v = -[J^o_f]^T \phi(t) \).

Unfortunately, in many cases, the Hessian matrix \( H^o_f \) is often difficult or impossible to compute. Even worse is that the convergence problem may arise when it is not definite positive. Hence, ref. [10] constructs the intrinsic Gauss-Newton algorithm by preserving the linear part and discarding the quadric item of Taylor series. In the next section, we will utilize the specific properties of the target tracking and image registration to propose a second-order minimization algorithm named by Liemanifold-SM.

It is well known that Lie-GN algorithm has high effective ness than the vector-GN algorithm by improving the iterative structure of parameters. Liemanifold-SM algorithm not only can avoid computing the complicated Hessian matrix but also can find back the Hessian matrix information discarded by Lie-GN algorithm to further improve the tracking performance.
3 PROJECTIVE TRACKING BASED ON SECOND-ORDER OPTIMIZATION ON LIE MANIFOLDS

3.1 Problem Statement

Suppose the camera is not be calibrated and the tracked object has a flat appearance. When the target is moving in the space, the relation between images can be described by the projective transformation. The projective transform group is the group of the matrices of the form $T = \begin{bmatrix} R & s \\ v & 1 \end{bmatrix}$, where $R$ is a $2 \times 2$ non-singular matrix, $s$ is a column vector for the translation and $(v, 1)^T$ is the projection of the line at infinity. We choose the scale factor to normalize the projective group $T$ such that the determinants of $T$ are equal to 1. Then the matrices $T$ belong to the special linear group $SL(3, R)$.

Let suppose the homogeneous coordinate of point $p$ be $(x, y, 1)^T$ and the embedding map in Euclidean space of $SL(3, R)$ be $\pi : t \rightarrow \pi(t)$. Define a group action from $SL(3, R)$ on $p : w : SL(3, R) \times p \rightarrow p$. The projective transformation is represented as follows

$$w(\pi(t))(p) = \frac{1}{a_{11}x + a_{12}y + a_{13}} \begin{bmatrix} a_{11}x + a_{12}y + a_{13} \\ a_{21}x + a_{22}y + a_{23} \\ a_{31}x + a_{32}y + a_{33} \end{bmatrix}$$

(5)

The algorithm initializes the tracker by assigning a rectangular region containing the target in the first frame as the initial template. Let $I(p)$ be the brightness value of the target and $I(wt(I(t)))$ be the intensity of the transformed target in the input image. The Jacobi matrix at the optimal transformation $t^*$, hence avoiding computing the Hessian matrix at the same time.

Let $f_p(t \circ \exp(\omega)) = I \circ t \circ \exp(\omega)(\omega) - I(\omega)$. From (3), we can draw

$$\left\| f_p(t \circ \exp(\omega)) \right\| = \left\| I \circ t \circ \exp(\omega)(\omega) - I(\omega) \right\|$$

$$= \left\| I(t(\omega)) - I(\omega) + \frac{1}{2} (J_{t'}(\omega) + J_{t'}(\omega)) \right\| + \mathcal{O}(1)$$

(8)

We pay attention to fact that when the images are aligned with the optimal spatial transformation in target tracking, the template and the warped image as well as their gradient should be very close to each other, i.e. $\nabla_p I \circ t' = \nabla_p I$. An efficient tracking algorithm will be constructed by utilizing this information to recover the information discarded with Gauss-Newton method by means of expanding the Jacobian matrix at the optimal transformation $t^*$.

The following is to compute the Jacobian matrix $J_{t'}(0)$ and $J_{t'}(v)$ corresponding to the derivative at 0 and $v$.

$$J_{t'}(0) = \frac{\partial I \circ t \circ \exp(\sum_{i=0}^{n} v_{e_i})}{\partial q}|_{q=0}$$

$$= \frac{\partial I \circ t(q)}{\partial q} \left|_{q=\pi(t)} \cdot \frac{\partial \pi(t(\omega))}{\partial \omega} \right|_{\omega=0}$$

$$= \nabla_p^T(I \circ t) \frac{\partial \pi(t(\omega))}{\partial \omega} \bigg|_{\omega=0}$$

(9)

For every line of the matrix $J_{t'}(0)$, $\left[ \nabla_p^T(I \circ t) \right]_{i \times 3}$ is corresponding to the spatial derivative of the current warped image using the projective transformation $t$; $\left[ J_{t'} \right]_{3 \times 9}$ is the Jacobian matrix for projective transformation (5), and $[e_x \ e_y \ e_z]$ is the Jacobian matrix where $\pi(e_i)$ is the matrix $e_i$ reshaped as a vector ( the entries are picked line per line ). The two Jacobians $J_{t'}$ and $e_x$ are constants to be computed once and for all while the Jacobian $J_{t'}(0)$ has to be computed at each iteration since it depends on the updated value of projective parameters.

However, the Jacobian matrix $J_{t'}(v)$ is complicated and usually depends on $t$. Hence, we do not directly compute $J_{t'}(v)$. If replacing the gradient of the optimally warped image $I \circ t' = I \circ t \circ \exp(\omega)$ by its equivalent gradient of the template image, we can get a simple linear approximation of $J_{t'}(v) \cdot v^*$ as follows

$$J_{t'}(v^*) \cdot v^* = \nabla_p^T(I \circ t') \cdot e_x \cdot v^*$$

(10)

Let $J_{t}$ be the following matrix

$$J_{t}$$
\[ J_i = -\frac{1}{2}(\nabla_p^T J + \nabla_p^T (I \circ t)) J \circ e_\pi \] (11)

Incorporating (11) into (8), we have
\[ f_p(t \circ \exp(\sigma^2 \pi)) = \| I \circ t(p) - I(p) + J \circ e_\pi + O \|^2 \] (12)

This cost function has a local or global minimum at \( v \)
\[ v^* = [J_i]^T (I \circ t(p) - I(p)) \] (13)

where \([J_i]^T\) is the pseudo-inverse of \( J_i \). Hence, the intrinsic iterative update is
\[ t = t \circ \exp(\sum_n^\pi v_i e_i) \]. (14)

### 3.3 Projective Tracking Process

To be self-closed for the paper, we present the complete target tracking process in which the efficient second-order minimization optimization algorithm is based on the Lie manifolds structure as follows:

1. **Pre-compute:**
   - Evaluate the Jacobian matrix \( J \circ e_\pi = \frac{\partial w(t, p)}{\partial t} \) and \( e(\pi) = [\pi(e_1) \cdots \pi(e_n)] \)
   - Define a template in the first frame and initialize the projective parameters \( t \) of template; input a new frame.
2. **Iterate:**
   - Crop a candidate target region of intensity \( I \circ t(p) \) in the new frame according to the parameters from previous iteration.
   - Compute the difference image between target image and template image: \( I \circ t(p) - I(p) \).
   - Compute the gradient of the input candidate target image: \( \nabla_p(I \circ t) \).
3. **Compute the Jacobian image:** \( J \circ e_\pi = \nabla_p^T (I \circ t) J \circ e_\pi \).
4. **Compute the parameter increment** \( v \) using equation (13).
5. **Choose the suitable parameter** \( \lambda \) and update the projective parameters: \( t \leftarrow t \circ \exp(\lambda v) \).
6. **If** \( \| v \| > \varepsilon \) **or iterations are below the predefined positive threshold, or the iterative number is less than the predefined value, return to step (2).**

Next frame:
7. **Read a new image then return to step (2), otherwise stop the tracking.**

### 4 EXPERIMENTAL RESULTS

To validate the feasibility and efficiency of our algorithm, we compare our Lie manifold-SM algorithm with Vector-GN algorithm in ref. [9] and Lie-GN algorithm in ref. [10]. All the algorithms are implemented in matlab and tested in the computer with Intel PIV 2.4GHZ and 512 Memory. Since the 8 parameters in the projective warp have different units, we compute the RMS (root-mean-square) error of the corresponding points between the template and target image rather than the RMS of parameters. In addition, it should be emphasized that neither preliminary image filtering nor multi-scale pyramid implementations nor other robust techniques have been used for this evaluation.
parameters that these perturbed points define (for each standard variance, we generated 100 randomly inputs). We say that an algorithm converged if the RMS error in the canonical point locations is less than 3.0 pixels after 15 iterations. We computed the percentage of times that each algorithm converged for each standard variance. The results are shown in Figure 3 that shows when the perturbation to the canonical point locations is less than about 3.0 pixels, all the three algorithms converge almost always. With the increase of the $\sigma$, the frequency of convergence for Vector-GN algorithm rapidly decreases. While $\sigma = 10$, the frequency of convergence for Vector-GN algorithm, Lie-GN algorithm and our Liemanifold-SM algorithm is 30%, 49% and 61% respectively. For 100 times experiments of $\sigma = 6$, all experiment test data are shown in Figure 4. Our Liemanifold-SM algorithm requires 8 iterations to coverage while Lie-GN requires 9 iterations and Vector-GN requires 15 iterations.

Experiment 2: We show experiment results for the typical image sequences with deformable targets tested using the three algorithms. The sequence is taken from a VR object movie, a sequence of one hundred still frames. The size of each frame is $512 \times 480$. The tracked target is the window of the house and the size of template is $52 \times 40$. The tracked region shows larger projective transformation. The Vector-GN algorithm, Lie-GN algorithm and Liemanifold-SM algorithm converge after 7, 5 and 4 iterations respectively. Figure 5 (a), 5 (b) and 5 (c) show some tracking results of them. The sequences after 80th frame have larger deformation. The Vector-GN algorithm cannot converge and the tracker slides off the tracking region. Lie-GN algorithms cannot lock the tracking region at 91st frame. However, our Liemanifold-SM algorithm can be implemented very well on all the frames. That our Liemanifold-SM algorithm not only improves the iterative structure but also avoids computing complicated Hessian matrix guarantees its efficiency and accuracy superiority than the other algorithms.

Fig.5: Tracking Results with Three Algorithms

5 CONCLUSION

Exploiting the differential geometry approaches, especially the geometric optimization techniques, to study the practical problem in the fields of nonlinear signal and computer vision is a remarkable direction recently. This paper constructed the tracking algorithm without computing the Hessian matrix based on the Lie manifold optimization. We compare our tracking algorithm with the classical vector space algorithm and the Gauss-Newton optimization algorithm based on the Lie algebra. The experimental results show that the accuracy and the convergent rate demonstrate some evident improvements.

ACKNOWLEDGEMENTS

This work is in part supported by the National Defense Science and Technology Creation Foundation of the Chinese Academy Sciences (Grant No.CXJJ-65).
REFERENCES