3D ROTATIONAL VIDEO STABILIZATION USING MANIFOLD OPTIMIZATION

Chao Jia and Brian L. Evans

Department of Electrical and Computer Engineering
The University of Texas at Austin, Austin, Texas, 78712 USA
Email: cjia@utexas.edu, bevans@ece.utexas.edu

ABSTRACT

We present a novel video stabilization method for cell phone cameras. Our video stabilization is based on a pure 3D rotation motion model, which can better capture the motion of the camera compared with 2D models. 3D camera rotation can be reliably captured by a gyroscope as commonly found on a smart phone or tablet. In this paper we directly smooth the sequence of camera rotation matrices for the video frames. Our contributions are (1) a smoothness metric for a sequence of 3D rotation matrices based on geodesic distance on a non-linear manifold, and (2) an efficient global motion smoothing algorithm using manifold optimization. Our smoothness metric better exploits the manifold structure of sequences of rotation matrices. Experimental results show that our video stabilization method outperforms state-of-the-art methods by generating more stable and visually pleasant videos.

***Index Terms*—** Video stabilization, manifold optimization, special orthogonal group

1. INTRODUCTION

Hand-held video cameras, such as in smart phones and tablets, are widely used to capture interesting or memorable moments conveniently at any time. Videos shot with hand-held cameras, however, often suffer from annoying jitter due to camera shake. Video stabilization aims at removing the unwanted jitter to generate visually stable and pleasant videos. Generally video stabilization consists of three major steps: (1) camera motion estimation, (2) camera motion smoothing and (3) frame synthesis. In this paper we focus on the second step.

We use a 3D rotational camera motion model for a calibrated camera with a known intrinsic matrix. Compared to 2D affine or projective motion models, 3D motion models can more accurately reflect the real camera perspective projection, and thus give more realistic motion smoothing and avoid image distortion in frame synthesis. We ignore 3D translation of the camera because (1) the unwanted jitter in videos are primarily caused by camera rotation, and (2) frame synthesis with 3D camera translation would need the depth value at every pixel, which is impossible to obtain. To estimate the 3D camera rotation we use a gyroscope that is available in many smart phones and tablets. Current gyroscopes in smart phones have very high precision and can return more reliable 3D camera rotation estimates compared to the estimates obtained from visual features in the video sequence, especially when there are many moving objects in the scene or it is difficult to track features points due to motion blur and illumination changes.

Under a 3D rotational model, camera motion for a video can be considered as a sequence of 3D rotation matrices. We formulate motion smoothing with a regularization term indicating the smoothness of the sequence of rotation matrices. Unlike traditional approaches, we exploit the manifold structure of the sequence of rotation matrices. The formulated problem is based on geodesic distance on the Riemannian manifold. Previous methods only exploit the properties on the manifold of the individual 3D rotation matrix $\mathbf{SO(3)}$ (Special Orthogonal Group), so they can only smooth the camera motion locally through low-pass filtering. We further consider the entire set of sequences of rotation matrices as a Riemannian manifold, so that we can model the motion smoothing problem globally and solve it optimally. We propose to use Netwon's algorithm on the manifold structure, which has much better convergence property than normal non-linear optimization algorithms in the Euclidean space. Experimental results show that our motion smoothing method outperforms state-of-the-art methods by generating more stable videos with less distortion.

2. RELATED WORK

Camera motion has been commonly modeled using 2D affine or projective approaches. Using full 3D models including both rotation and translation for calibrated cameras was first proposed in [1] and further discussed in [2]. In both papers complicated approximations are used in frame synthesis to handle the problem of missing depth values. In [3] and [4] pure 3D rotational models were shown to generate high-quality results while only needing homography-based warping in frame synthesis.

Gyroscopes and other inertial measurement sensors have been widely used in robotic localization problems together with visual measurements. However, they were not used in video stabilization to replace the feature-based motion estimation until they became accurate enough and widely available in cell phones recently[4, 5].

Motion smoothing methods using 2D models are based on Euclidean distance. 2D camera motion can be smoothed using local methods such as Gaussian-kernel low-pass filtering [6], global methods such as L_1-based regularization [7], and real-time methods such as Kalman filtering [8]. 3D rotation smoothing has been implemented locally by low-pass filtering based on either Euclidean distance [4] or geodesic distance on the manifold $\mathbf{SO(3)}$ [2, 5]. Although $\mathbf{SO(3)}$ has additional applications in computer vision, medical imaging and robotics [9], we have not found any previous work considering the sequence of 3D rotation matrices as a whole. In this paper we directly exploit the manifold structure of sequences of rotation matrices so that we can formulate 3D rotation smoothing as a regression problem.

This research was supported by gift funding from TI.

3. GEODESIC DISTANCE AND VIDEO STABILIZATION

All of the 3×3 rotation matrices constitute an embedded Riemannian submanifold of the set of all 3×3 real matrices. In group theory this manifold is known as special orthogonal group $\mathbf{SO(3)}$, in which any element $\mathbf{R}$ satisfies the orthonormality constraint $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$. A natural extension of Euclidean distance in Euclidean space to the Riemannian manifold $\mathbf{SO(3)}$ is the geodesic distance

$$d_g(\mathbf{R}_m, \mathbf{R}_n) = ||\mathrm{logm}(\mathbf{R}'_m\mathbf{R}_n)||_F, \tag{1}$$

where $\mathrm{logm}(\cdot)$ is the matrix logarithm operator and $||\cdot||_F$ is the Frobenius norm of a matrix. In fact, $\mathrm{logm}(\mathbf{R}'_m\mathbf{R}_n)$ is a skew-symmetric matrix representing a tangent vector in the tangent space $T_{\mathbf{R}_m}\mathbf{SO(3)}$ that indicates the non-normalized direction from $\mathbf{R}_m$ to $\mathbf{R}_n$ on $\mathbf{SO(3)}$. Usually we also write $\mathrm{logm}(\mathbf{R}'_m\mathbf{R}_n)$ as $\log_{\mathbf{R}_m}\mathbf{R}_n$ and call it a logarithmic mapping. Inversely, given any tangent vector $\xi \in T_{\mathbf{R}_m}\mathbf{SO(3)}$, we can define $\exp_{\mathbf{R}_m}\xi = \mathbf{R}_m\mathrm{expm}(\xi)$, where $\mathrm{expm}(\cdot)$ is the matrix exponential operator. Here, $\exp_{\mathbf{R}_m}\xi$ is called an exponential mapping and is used to move $\mathbf{R}_m$ along the direction defined by ξ on $\mathbf{SO(3)}$.

For each video sequence, we can obtain a sequence of 3D rotation matrices corresponding to all of the frames from the gyroscope readings. Next we consider the sequence of 3D rotation matrices as a whole and exploit the properties of the Riemannian manifold constituted by these sequences.

Assume the sequence of 3D camera rotation for any video sequence with N frames can be represented by

$$\mathbf{x} = [\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N]^{\mathrm{T}}. \tag{2}$$

All of the possible rotation matrix sequences with N elements constitute a manifold $\mathcal{M}_R$ with dimension $3N$. Indeed, this manifold is a Cartesian product of N $\mathbf{SO(3)}$ manifolds.

$$\mathcal{M}_R = \mathbf{SO(3)} \times \mathbf{SO(3)} \times \ldots \times \mathbf{SO(3)}. \tag{3}$$

The manifold $\mathcal{M}_R$ is also an embedded Riemannian submanifold of $3N \times 3$ real matrices ($\simeq \mathbb{R}^{9N}$). Furthermore, for any $\mathbf{x} \in \mathcal{M}_R$, the tangent space $T_{\mathbf{x}}\mathcal{M}_R$ at $\mathbf{x}$ can be represented by

$$[\Omega_1, \Omega_2, \ldots, \Omega_N]^{\mathrm{T}}, \tag{4}$$

where $\{\Omega_n\}$ are real skew-symmetric matrices. In other words, the tangent vectors and corresponding exponential (and logarithmic) mapping are still separable as the elements in the manifold of rotation matrix sequences. This fact makes the gradient-related optimization algorithms easy to implement.

The goal of video stabilization is to remove visible jitter and make the camera motion trajectory change smoothly. Given the manifold structure of $\mathbf{SO(3)}$, it is natural to define the smoothness of a rotation matrix sequence as the sum of geodesic distances between adjacent rotation matrices. At the same time, we need to guarantee that the smoothed camera motion trajectory does not deviate from the original trajectory too much. As a result, we formulate the video stabilization problem as

$$\min_{\{\mathbf{R}_n^{new}\}} \sum_{n=1}^{N} \frac{1}{2} d_g^2(\mathbf{R}_n^{old}, \mathbf{R}_n^{new}) + \alpha \sum_{n=1}^{N-1} \frac{1}{2} d_g^2(\mathbf{R}_n^{new}, \mathbf{R}_{n+1}^{new}), \tag{5}$$

where $\{\mathbf{R}_n^{new}\}$ is the sequence of stabilized rotation matrices, $\{\mathbf{R}_n^{old}\}$ is the original sequence of rotation matrices, α is the weighting parameter controlling the smoothness of the stabilized trajectory. Note that although the objective function is derived based on the geodesic distance between elements in $\mathbf{SO(3)}$, it is defined on the rotation matrix sequence manifold $\mathcal{M}_R$. For brevity, we use $\mathbf{x} \in \mathcal{M}_R$ to represent the rotation matrix sequence $\{\mathbf{R}_n^{new}\}$ in form of (2) and write the objective function to minimize as $f(\mathbf{x})$.

4. SOLVING VIDEO STABILIZATION USING GRADIENT-RELATED METHODS

The formulated problem in (5) is equivalent to an unconstrained quadratic programming problem in Euclidean space. In Euclidean space, such problems have closed-from solutions; however, on non-linear manifolds we have to use iterative algorithms. Gradient-related iterative algorithms are widely used in optimization for manifolds as for Euclidean space [10]. For any element $\mathbf{x}$ in the manifold of rotation matrix sequence $\mathcal{M}_R$, given any tangent vector $\xi_{\mathbf{x}} \in T_{\mathbf{x}}\mathcal{M}_R$, we can move $\mathbf{x}$ along the direction defined by $\xi_{\mathbf{x}}$ using the exponential mapping $\exp_{\mathbf{x}}\xi_{\mathbf{x}}$. Note that given the separability property of the tangent vectors the exponential mapping can also be implemented separately for different rotation matrices in the sequence. If $\xi_{\mathbf{x}}$ is a descent direction related to the gradient of the objective function at $\mathbf{x}$, then we have the gradient-related algorithm on the manifold $\mathcal{M}_R$. In fact, similar convergence results of gradient-related algorithms has been extended from the Euclidean space to general manifolds [10]. In this paper we investigate two popular gradient-related descent directions: steepest gradient descent and Newton's method.

4.1. Steepest Gradient Descent

The descent direction of steepest gradient descent algorithm is simply the opposite of the gradient $\mathrm{grad}f(\mathbf{x})$. To compute the gradient we first rewrite the objective function as

$$f(\mathbf{x}) = \sum_{n=1}^{N} g_n(\mathbf{x}) + \alpha \sum_{n=1}^{N-1} h_n(\mathbf{x}), \tag{6}$$

where $g_n(\mathbf{x}) = \frac{1}{2}d_g^2(\mathbf{R}_n^{old}, \mathbf{R}_n^{new})$ and $h_n(\mathbf{x}) = \frac{1}{2}d_g^2(\mathbf{R}_n^{new}, \mathbf{R}_{n+1}^{new})$. Note that $\mathbf{R}_n^{new}$ is one of the 3×3 rotation matrix in $\mathbf{x}$. For brevity we define $\mathbf{R}_n^{new} = A_n\mathbf{x}$, where A_n is a $3 \times 3N$ matrix that is used to extract $\mathbf{R}_n^{new}$ from $\mathbf{x}$. Similarly we can map $\mathbf{R}_n^{new}$ back to its corresponding location in $\mathbf{x}$ by $A_n^{\mathrm{T}}\mathbf{R}_n^{new}$.

If we consider $\frac{1}{2}d_g^2(\mathbf{R}_n^{old}, \mathbf{R}_n^{new})$ as a function of $\mathbf{R}_n^{new}$, it has been proven [11] that

$$\mathrm{grad}\ \frac{1}{2}d_g^2(\mathbf{R}_n^{old}, \mathbf{R}_n^{new}) = -\log_{\mathbf{R}_n^{new}}\mathbf{R}_n^{old}. \tag{7}$$

Given the separability feature of $\mathbf{x}$, we can further obtain

$$\begin{cases} \mathrm{grad}\ g_n(\mathbf{x}) = -A_n^{\mathrm{T}}\log_{A_n\mathbf{x}}\mathbf{R}_n^{old} \\ \mathrm{grad}\ h_n(\mathbf{x}) = -A_n^{\mathrm{T}}\log_{A_n\mathbf{x}}A_{n+1}\mathbf{x} - A_{n+1}^{\mathrm{T}}\log_{A_{n+1}\mathbf{x}}A_n\mathbf{x}. \end{cases} \tag{8}$$

Using linearity of the gradient, we can obtain

$$\begin{aligned} \mathrm{grad}\ f(\mathbf{x}) = -A_1^{\mathrm{T}}(\log_{A_1\mathbf{x}}\mathbf{R}_1^{old} + \log_{A_1\mathbf{x}}A_2\mathbf{x}) \\ - \sum_{n=2}^{N-1} A_i^{\mathrm{T}}(\log_{A_n\mathbf{x}}\mathbf{R}_n^{old} + \log_{A_n\mathbf{x}}A_{n+1}\mathbf{x} + \log_{A_n\mathbf{x}}A_{n-1}\mathbf{x}) \\ -A_N^{\mathrm{T}}(\log_{A_N\mathbf{x}}\mathbf{R}_N^{old} + \log_{A_N\mathbf{x}}A_{N-1}\mathbf{x}). \end{aligned} \tag{9}$$

Equation (9) clearly shows the decomposition of grad $f(\mathbf{x})$ into N skew symmetric matrices corresponding to the N rotation matrices in $\mathbf{x}$. Given the direction, we can use exponential mapping to update $\mathbf{x}$ in each iteration with a proper step size to guarantee descent in the value of objective function. In this paper we choose the step size using the Armijo rule [12].

In Euclidean space the convergence rate of steepest gradient descent is strongly affected by the eigenvalues of the Hessian matrix of the objective function Hess $f(\mathbf{x})$. This property also holds for non-linear manifolds [10]. In fact we can check that the Hessian matrix of the given objective function is ill-conditioned (the largest eigenvalue is much larger than the smallest eigenvalue). Therefore, the steepest gradient descent method converges only sublinearly.

4.2. Newton's Method

Newton's method has been proven to converge locally quadratically to the optimal solution for both Euclidean space and non-linear manifolds. Newton's method needs calculating the Hessian Hess $f(\mathbf{x})$. To calculate the Hessian on manifolds is a very difficult task. We start to derive the Hessian of the proposed objective function from the following lemma in [13].

Lemma 1. *Consider the geodesic distance function* $\phi_{\mathbf{Q}}(\mathbf{P}) = d_g^2(\mathbf{P}, \mathbf{Q})$*, where* $\mathbf{P}, \mathbf{Q} \in \mathbf{SO(3)}$*. Let* $r = d_g(\mathbf{P}, \mathbf{Q})$ *be the geodesic distance. Let* $\gamma(t) : [0, r] \to \mathbf{SO(3)}$ *denote the unit speed geodesic connecting* $\mathbf{Q}$ *to* $\mathbf{P}$*.* $\forall \xi_{\mathbf{P}}, \eta_{\mathbf{P}} \in T_{\mathbf{P}}\mathbf{SO(3)}$*, we have the Hessian operator*

$$\text{Hess}\ \phi_{\mathbf{Q}}(\mathbf{P})(\xi_{\mathbf{P}}, \eta_{\mathbf{P}}) = \langle \xi_{\mathbf{P}}^{\|}, \eta_{\mathbf{P}}^{\|} \rangle + \frac{r}{\tan(r/2)} \langle \xi_{\mathbf{P}}^{\perp}, \eta_{\mathbf{P}}^{\perp} \rangle, \qquad (10)$$

where $\|$ *and* $\perp$ *signs denote parallel and perpendicular orthogonal components of the tangent vector with respect to* $\dot{\gamma}(r)$*. Here* $\dot{\gamma}(r) \in T_{\mathbf{P}}\mathbf{SO(3)}$ *is the parrallel translation of* $\dot{\gamma}(0) = \log_{\mathbf{Q}} \mathbf{P}$ *along the geodesic from* $\mathbf{Q}$ *to* $\mathbf{P}$*.*

Given Lemma 1 and any orthonormal basis $\{E^i\}_{i=1,2,3}$ of $T_{\mathbf{P}}\mathbf{SO(3)}$ we can compute the matrix representation of the Hessian operator by computing its result on every pair of basis tangent vectors. Lemma 1 gives us a way to compute the Hessian matrix when the objective function is the geodesic distance defined on $\mathbf{SO(3)}$. In our proposed problem we need to find the Hessian for $g_n(\mathbf{x})$ and $h_n(\mathbf{x})$, which are defined on the manifold $\mathcal{M}_R$ of rotation matrix sequences. Note that due to the separability feature of the tangent vectors of $\mathcal{M}_R$, we can always find an orthonormal basis $\{E_n^i\}_{i=1,2,3;n=1,\ldots,N}$ of $T_{\mathbf{x}}\mathcal{M}_R$, where only $A_n E_n^i$ is non-zero and it is equal to the basis vector E^i defined for $T_{A_n\mathbf{x}}\mathbf{SO(3)}$. In other words, the orthonormal basis of $T_{\mathbf{x}}\mathcal{M}_R$ can be represented by N subgroups and each subgroup corresponds to one particular rotation matrix in the entire sequence. We propose the following theorem:

Theorem 1. *Given the decomposed objective functions defined in equation (6) and an orthonormal basis of* $T_{\mathbf{x}}\mathcal{M}_R$ *in form of* $\{E_n^i\}$*, we have*

$$\begin{cases} \text{Hess}\ g_n(\mathbf{x})(E_n^i, E_n^j) = \text{Hess}\ \phi_{\mathbf{R}_n^{old}}(A_n\mathbf{x})(E^i, E^j) \\ \text{Hess}\ g_n(\mathbf{x})(E_m^i, E_l^j) = 0, \text{if } m \neq n \text{ or } l \neq n \end{cases} \qquad (11)$$

$$\begin{cases} \text{Hess}\ h_n(\mathbf{x})(E_n^i, E_n^j) = \text{Hess}\ \phi_{A_{n+1}\mathbf{x}}(A_n\mathbf{x})(E^i, E^j) \\ \text{Hess}\ h_n(\mathbf{x})(E_{n+1}^i, E_{n+1}^j) = \text{Hess}\ \phi_{A_n\mathbf{x}}(A_{n+1}\mathbf{x})(E^i, E^j) \\ \text{Hess}\ h_n(\mathbf{x})(E_n^i, E_{n+1}^j) = -\text{Hess}\ \phi_{A_{n+1}\mathbf{x}}(A_n\mathbf{x})(E^i, E^j) \\ \text{Hess}\ h_n(\mathbf{x})(E_{n+1}^i, E_n^j) = -\text{Hess}\ \phi_{A_n\mathbf{x}}(A_{n+1}\mathbf{x})(E^i, E^j) \\ \text{Hess}\ h_n(\mathbf{x})(E_m^i, E_l^j) = 0, \text{if } m \neq n, n+1 \text{ or } l \neq n, n+1 \end{cases} \qquad (12)$$

The computation on the right hand side of the equations has been defined in Lemma 1.

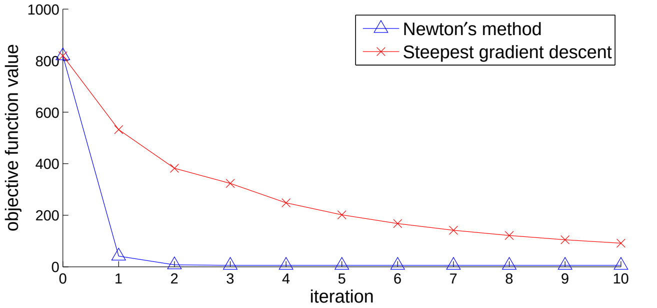


Fig. 1. Convergence of the gradient-related algorithms in video stabilization.

Due to paper length constraints, we only sketch the proof here. The proof is established by exploiting the independence among different components in the rotation matrix sequence and the definition of the Hessian operator based on Levi-Civita connection [14]. Using Theorem 1 and linearity of the Hessian we can obtain a $3N \times 3N$ matrix representation H of Hess $f(\mathbf{x})$ for a given orthonormal basis $\{E_n^i\}$. To compute the direction in Newton's method, we first compute grad $f(\mathbf{x})$ and then represent it as a vector v under the orthonormal basis $\{E_n^i\}$. Then we just need to solve the linear system $H \cdot u = -v$ and the direction is represented by the vector u under the same basis. Given the update direction we still use the Armijo rule to select the step size.

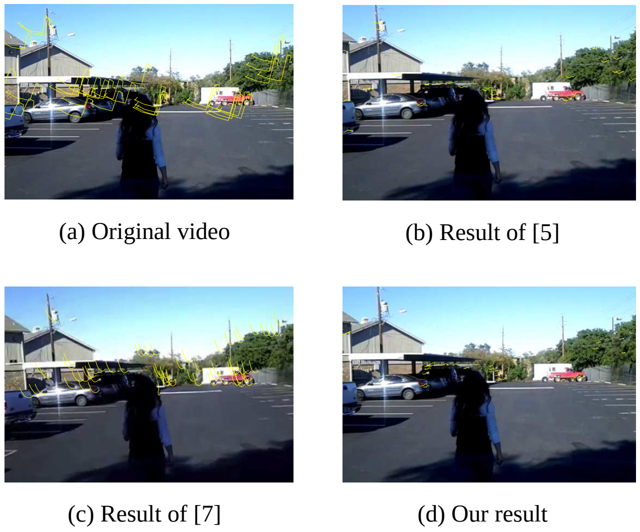
5. EXPERIMENTAL RESULTS

We first compare the convergence rate of steepest gradient descent method and Newton's method in solving the formulated problem. In the experiment we try to smooth a sequence of 478 3D rotation matrices (478 frames) with $\alpha = 1000$. Figure 1 shows the values of the objective function in 10 iterations. Newton's method successfully converges in just 2 iterations. Each iteration of the Newton's method takes 2.93 seconds on a 2.3GHz Intel i5 processor machine with MATLAB implementation (without parallel processing). We find that the number of iterations needed before convergence is not affected by the total number of frames in the video, so the convegence time increases gracefully (linearly) with the increase in the number of frames.

We use the proposed motion smoothing method in video stabilization and compare the results with two state-of-the-art methods: (1) L_1 regularization on 2D affine models [7] (used in the YouTube video editor) and (2) Low-pass filtering on 3D rotation model using Hamming window [5]. The 2D motion is estimated from tracked feature points, while the 3D rotations is directly obtained by integrating the gyroscope readings. None of the methods is full-frame video stabilization so we need to crop the stabilized outputs. The original frame size of the videos is 720×480 and the stabilized video from the YouTube video editor [7] has been automatically cropped into 540×360. So in comparison we use the same size to crop the results of our method and our implementation of [5] to make the frame size consistent. However, we need to mention that our method can generate larger-size cropped videos without unknown boundaries. In the experiments we fix the smoothness parameter as $\alpha = 100$.

First, we test different methods on a video shot by a walking forward person. In Figure 2 we detect Harris corner points in a certain frame and track them for ten frames. We show the starting frame with yellow curves indicating the tracks of the feature points in the following 10 frames. For a stabilized video the tracks should be

Fig. 2. Stabilization comparison for a video shot by a walking forward person.



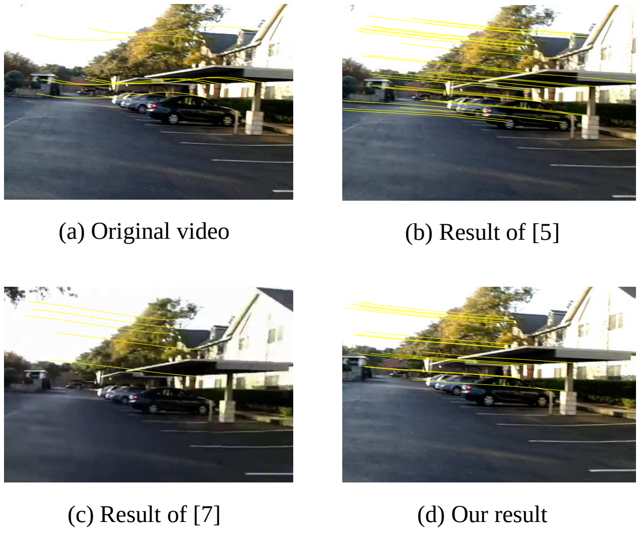
(a) Original video

(b) Result of [5]

(c) Result of [7]

(d) Our result

Fig. 3. Stabilization comparison for a video shot while panning the camera.



(a) Original video

(b) Result of [5]

(c) Result of [7]

(d) Our result

very short since the camera is always facing forward in spite of jitter caused by hand shake. The 2D L_1-regularization method [7] can smooth and shorten the tracks compared to the original video, but the feature points are still moving up and down. 3D local low-pass filtering method [5] and our algorithm can keep the feature points very steady and our result is slightly better than [5]. Note that we detect the feature points independently in the four videos so the location and number of the feature points are different.

Next we take a test on a video shot while panning the camera. Video stabilization should only remove the unwanted jitter while keeping the panning motion of the camera. In Figure 3 we do the same kind of test as in Figure 2. All of the three methods successfully smooth the tracks of the feature points. The tracks in the result of [7] are not as straight as those in the result of [5] and our result if we zoom in the results (notice the two ends of the tracks).

The stabilization results are best viewed in video form. Please see the video examples on the Web page of our paper [15]. In the local comparison (10 frame duration) in Figure 2 and Figure 3 the local smoothing method [5] performs similarly to our method. However, our method works better globally. In addition, the weighted average computation on rotation matrices based on geodesic distance cannot be solved analytically and needs iterative algorithms. So the local low-pass filtering method in [5] needs to run iterative algorithms for each frame and thus takes longer than our method, especially when the Hamming window size is large. Some other works tried to solve the rotation averaging approximately [2] or just based on Euclidean distance to increase the processing speed but with sacrificing stabilization quality. In the two video examples features are easy to track since there is very little motion blur in the frames. However, when the videos are shot in low light condition the visual-based motion estimation used in [7] will fail sometimes while the 3D rotational video stabilization using gyroscopes is not affected.

6. CONCLUSIONS

In this paper we propose a novel video stabilization method using a 3D rotational camera motion model. We exploit the manifold structure of not only the 3D rotation matrices, but also the sequences of 3D rotation matrices. We formulate the global motion smoothing as a regularization problem based on geodesic distance and present an efficient Newton's algorithm to solve the problem on the proposed manifold. The 3D camera rotation for each frame is obtained reliably using gyroscopes that are equipped in most smart phones and tablets. We have demonstrated in experiments that our algorithm is very fast and can generate better video stabilization results than state-of-the-art methods.

7. REFERENCES

[1] C. Buehler, M. Bosse, and L. McMillan, "Non-metric image-based rendering for video stabilization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Dec. 2001, pp. 609–614.

[2] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization," *ACM Trans. on Graphics*, vol. 28, no. 3, 2009.

[3] C. Morimoto and R. Chellappa, "Fast 3D stabilization and mosaic construction," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 1997.

[4] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes," Tech. Rep., Stanford University, Mar. 2011.

[5] G. Hanning, N. Forslöw, P.-E. Forssén, E. Ringaby, D. Törnqvist, and J. Callmer, "Stabilizing cell phone video using inertial measurement sensors," in *Proc. IEEE Intl. Workshop on Mobile Vision*, Nov. 2011.

[6] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE*

Trans. on Pattern Analysis and Machine Intelligence, vol. 28, July 2006.

[7] M. Grundmann, V. Kwatra, and Ifran Essa, "Auto-directed video stabilization with robust L1 optimal camera paths," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2011.

[8] A. Litvin, J. Konrad, and W. Karl, "Probabilistic video stabilization using Kalman filtering and mosaicking," *Proc. IS&T/SPIE Symposium on Electronic Imaging, Image and Video Comm.and Proc.*, pp. 663–674, 2003.

[9] V. Govindu, "Lie-algebraic avearaging for globally consistent motion estimation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2004.

[10] P.-A. Absil, R. Mahony, and R. Sepulchrer, *Optimization Algorithm on Matrix Manifolds,* Princeton University Press, 2008.

[11] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Comm. Pure Appl. Math*, vol. 30, pp. 509–541, 1977.

[12] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, 1999.

[13] R. Ferreira, J. Xavier, J.P. Costeira, and V. Barroso, "Newton method for Riemannian centroid computation in naturally reductive homogeneous spaces," in *Proc. IEEE Intl. Conference on Acoustics, Speech and Signal Processing*, May 2006.

[14] J. Gallier, *Notes on Differential Geometry and Lie Groups*, University of Pennsylvannia, 2012.

[15] C. Jia and B. L. Evans, "Video demonstrations for 3D rotational video stabilization using manifold optimization," `http://www.ece.utexas.edu/~bevans/papers/2013/stabilization/`.