A motion tracking method by combining the IMU and camera in mobile devices

Conference Paper · November 2016

DOI: 10.1109/ICSensT.2016.7796235

Wei Fang
10 PUBLICATIONS **33** CITATIONS
SEE PROFILE

Lianyu Zheng
Beihang University (BUAA)
74 PUBLICATIONS **637** CITATIONS
SEE PROFILE

A motion tracking method by combining the IMU and camera in mobile devices

Wei Fang, Lianyu Zheng*
School of Mechanical Engineering and Automation
Beihang University
Beijing China
{wfang, lyzheng}@buaa.edu.cn

Huanjun Deng
Beijing Baofengmojing Technologies Co., Ltd
Beijing China
denghuanjun@baofeng.com

Abstract—**In order to track the localizations of mobile devices in an unknown environment, this paper presents an architecture combining a monocular camera and an inertial measurement unit (IMU) in ubiquitous mobile devices. The IMU module provides acceleration and angular velocity with high-frequency, but the IMU-based motion tracking is more inclined to collapse due to the drift integration. While the vision-based motion tracking can provide higher accuracy, but it cannot work in the environment with weak texture or dynamic scenes. Based on the fusion of the IMU and monocular camera, this paper proposed a loosely couple method in the error-state Extended Kalman Filter framework. With the combination of the advantages the monocular camera and IMU, the proposed method can achieve real-time ego-motion estimation in resource-constrained mobile devices. Finally, the validity of the proposed motion tracking method is evaluated by experiments.**

Keywords-motion tracking; Kalman Filter; sensor fusion, visual odometry, IMU

I. Introduction

In recent years, motion tracking is a fundamental problem in the research area of augmented reality [1], self-driving [2], unmanned aerial vehicles [3], production planning [4] and several others. It can provide the position, orientation and velocity of the object in an unknown environment. With these information, the autonomous equipment can percept the current pose estimation and do the decision-making for the next step. This technology is becoming ubiquitous in both industrial application and our daily lives. Typically, a motion tracking system includes multiple sensors that are used to capture information about the system's motion and its surroundings, and a computer is used to process the sensor information for producing estimates about the system's position and orientation.

With the developments of the lightweight and cheap micro-electro-mechanical system (MEMS) inertial sensors, the smartphones and other similar mobile devices are almost equipped with low-cost embedded IMU sensors. As a primary motion capture sensor, which can provide the position, orientation, and velocity through time integration of linear acceleration and angular velocity. Nevertheless, low cost IMUs are affected by the systematic error, the position and the orientation must be corrected periodically. This correction can be provided by complementary sensors, such as a global positioning system (GPS), but it cannot work in the GPS-denied environment [5].

Vision-based motion tracking is another technique that estimates the ego-motion by using single or multiple camera, and the term visual odometry is first proposed by Nister et al. [6], which operates by incrementally estimating the pose by examination of the changes that motion induces on the images. It has been demonstrated that the motion tracking from visual odometry can provide more accurate trajectory estimates, with relative position error range from 0.1 to 2% [7]. The main vision-based motion tracking techniques either use dense motion algorithm, known as optical flow [8] or feature-based tracking method [9]. However, these approaches suffer from navigating in poor feature environments and scenes that contain complex backgrounds or dynamic objects, this makes tracking of salient image features un-tractable for motion estimation. Furthermore, consecutive frames should be captured by ensuring that they have sufficient scene overlap. In order to improve the robustness of the motion tracking, other sensors should be taken into consideration for assist. Therefore, the camera, equipped with an IMU in almost every mobile device, provides one alternative that can be used for robust motion tracking applications. The combination of a camera and an IMU in mobile devices for motion tracking is proposed in this paper.

In order to make the vision and inertial sensors compensate with each other, the sensor fusion pay a key role in our motion tracking method. The inertial data are updated with high-frequency, such as 100 Hz or more, and the camera trajectory can be recovered from the captured images with about 25Hz. The work in [10] shows the fusion of IMU and vision for localization, and they compare the performance of an Extended Kalman Filter (EKF) and an Unscented Kalman Filter (UKF). The authors state that UKF approach yields better results at the cost of calculate power. In our mobile devices, the speed of the algorithm is given high priority due to their limited computing resources. Therefore, the EKF framework is chosen for our fusion estimation of the IMU and vision.

Within the EKF framework, the error-state is always small, meaning that the second order products are negligible, which makes the computation of the Jacobians very easy and fast. In addition, the error dynamics are slow for the larger-signal dynamics, it also means that the corrections are at a lower rate

978-1-5090-0795-0/15/$31.00 ©2016

than the prediction [11]. Therefore, given prior consideration for the limited computation in current mobile devices, this paper proposed a loosely couple method within the error-state Extended Kalman Filter framework. The propagation of the state vector and covariance is derived from the IMU, and the update process occurred when the image arrived through epipolar geometry. The experimental results demonstrate that the proposed method can run in real-time in the resource constrained mobile devices.

The following sections are organized as follows. Section II introduces the IMU driven system for state propagation. In Section III, the vision-based motion tracking is described, which can provide the measurement for the EKF framework. Section IV illustrates the measurement update for IMU-vision motion tracking fusion. In section V, the performance of the proposed method is demonstrated by the experiments. Finally, the conclusion and the future work of the study is summarized in section VI.

II. IMU Driven System

The main purpose of our proposed method is to estimate the position and orientation of the mobile device in unknown environments. The relationship of different coordinates is shown in Fig. 1, where the IMU frame $\{I\}$ and the camera frame $\{C\}$ are rigidly mounted, and the world frame is depicted as $\{W\}$. The quaternion and position pair $\{q_w^i, p_w^i\}$ denotes the transformation of the IMU in global frame, while the $\{q_w^c, p_w^c\}$ represents the transformation of the camera with respect to the global frame. The pair $\{q_i^c, p_i^c\}$ denotes the attitude and position of the IMU in camera frame, which is a fixed value and can be calibrated by the method [12] in advance.

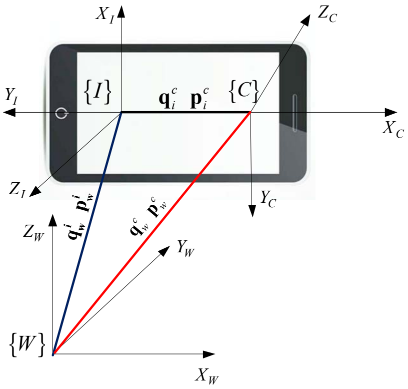

X_I
Z_C
Y_I
$\{I\}$
$\mathbf{q}_i^c$ $\mathbf{p}_i^c$
$\{C\}$
X_C
Z_I
Y_C
Z_W
$\mathbf{q}_w^i$ $\mathbf{p}_w^i$
$\mathbf{q}_w^c$ $\mathbf{p}_w^c$
Y_W
$\{W\}$
X_W

Fig. 1. IMU-Camera coordinates in mobile devices

A. IMU measurement model

The measurements of the IMU contain a certain bias b and white Gaussian noise n. Thus, the real angular velocity ω and the real acceleration a are related with gyroscope and accelerometer measurements can be obtained, respectively.

$$\omega_m = \omega + b_\omega + n_\omega \qquad a_m = a + b_a + n_a \tag{1}$$

The subscript m denotes the measured value, and dynamics of the non-static bias b are modeled as a random process.

B. State representation

The state vector used to describe the IMU comprises its position, orientation, velocity and the biases of the the gyroscope and accelerometer. To make the posture fusion from the IMU and vision, the rotation and translation from the IMU to the camera frame are also included in the filter state. Thus, the state vector X can be obtained:

$$X = \left\{ p_w^{i\,T} \; v_w^{i\,T} \; q_w^{i\,T} \; b_\omega^{\;T} \; b_a^T \; p_i^{cT} \; q_i^{cT} \right\} \tag{2}$$

Then, the data-driven dynamic model can be represented by the following differential equations:

$$\begin{gathered} \dot{p}_w^i = v_w^i, \; \dot{v}_w^i = C_{\left(q_w^i\right)}^T a - g, \; \dot{q}_w^i = \frac{1}{2}\Omega(\omega) q_w^i \\ \dot{b}_\omega = n_{b\omega}, \; \dot{b}_a = n_{ba}, \; \dot{p}_i^c = 0, \; \dot{q}_i^c = 0 \end{gathered} \tag{3}$$

Where, $C_{\left(q_w^i\right)}^T$ is the rotational matrix corresponding to the quaternion q_w^i, and g is the gravity vector in the world frame $\{W\}$. $\Omega(\omega)$ is the quaternion multiplication matrix of ω, and $\Omega(\omega) = \begin{bmatrix} -\lfloor \omega \rfloor_\times & \omega^T \\ -\omega^T & 0 \end{bmatrix}$, $\lfloor \omega \rfloor_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$ is the skew-symmetric matrix. Assuming $\bar{q} = (q_0, q)^T$ is a unit quaternion and its corresponding rotational matrix is represented as $C_{\bar{q}}$. These two orientation representations can be related as below:

$$C_{\bar{q}} = (2q_0^{\,2} - 1)I_3 - 2q_0 \lfloor q \rfloor_\times + 2qq^T \tag{4}$$

Since the mean of the noise is assumed as zero, the IMU nominal-state kinematics can be obtained by taken expectations of the IMU state propagation:

$$\begin{gathered} \dot{\hat{p}}_w^i = \hat{v}_w^i, \; \dot{\hat{v}}_w^i = C_{\left(\hat{q}_w^i\right)}^T \left(a_m - \hat{b}_a\right) - g \\ \dot{\hat{q}}_w^i = \frac{1}{2}\Omega\left(\omega_m - \hat{b}_\omega\right)\hat{q}_w^i \\ \dot{\hat{b}}_\omega = 0, \; \dot{\hat{b}}_a = 0, \; \dot{\hat{p}}_i^c = 0, \; \dot{\hat{q}}_i^c = 0 \end{gathered} \tag{5}$$

C. Error state representation

In order to minimize the dimension of the filter state vector and achieve the linearization, the 21-elements error state vector can be acquired:

$$\tilde{x} = \left\{ \Delta p_w^{i\,T} \;\; \Delta v_w^{i\,T} \;\; \delta\theta_w^{i\,T} \;\; \Delta b_\omega^{\;T} \;\; \Delta b_a^{\;T} \;\; \Delta p_i^{cT} \;\; \delta\theta_i^{cT} \right\} \tag{6}$$

Given the error state filter formulation, the relationship between the true state x, nominal state $\hat{x}$, and error state $\tilde{x}$ is:

$$x = \hat{x} + \tilde{x} \tag{7}$$

The error quaternions can be defined as follow:

$$\begin{gathered} \delta q_w^i = q_w^i \otimes \hat{q}_w^{i\,-1} \approx [\frac{1}{2}\delta\theta_w^{i\,T} \;\; 1]^T \\ \delta q_i^c = q_i^c \otimes \hat{q}_i^{c-1} \approx [\frac{1}{2}\delta\theta_i^{cT} \;\; 1]^T \end{gathered} \tag{8}$$

Then, with $\hat{\omega}=\omega_m-\hat{b}_\omega$ and $\hat{a}=a_m-\hat{b}_a$, the differential equations for the continuous time error state can be obtained:

$$\begin{gathered}
\Delta\dot{p}_w^i=\Delta v_w^i \\
\Delta\dot{v}_w^i=-C_{(\hat{q}_w^i)}^T\lfloor\hat{a}\rfloor_\times\delta\theta-C_{(\hat{q}_w^i)}^T\Delta b_a-C_{(\hat{q}_w^i)}^T n_a \\
\delta\dot{\theta}_w^i=-\lfloor\hat{\omega}\rfloor_\times\delta\theta-\Delta b_\omega-n_\omega \\
\Delta\dot{b}_\omega=n_{b_\omega},\ \ \Delta\dot{b}_a=n_{b_a},\ \ \Delta\dot{p}_i^c=0,\ \ \Delta\dot{\theta}_i^c=0
\end{gathered} \tag{9}$$

D. Filter propagation

At the filter prediction stage, the inertial measurements for state propagation are obtained from an IMU in discrete form, thus the signals from gyroscope and accelerometer are assumed to sample with a certain time interval, and the nominal state can be obtained with the numerical integration of the 4th Runge-Kutta method. By stacking the differential equations for error state, the linearized continuous time error state equation can be given:

$$\dot{\tilde{x}}=F_c\tilde{x}+G_c n \tag{10}$$

Where the noise vector $n=[n_a^T,n_{b_a}^T,n_\omega^T,n_{b_\omega}^T]^T$, and F_d can be obtained by digitizing F_c by the Taylor series:

$$F_d=\exp\left(F_c\Delta t\right)=\mathbf{I_d}+F_c\Delta t+\frac{1}{2}F_c^2\Delta t^2+\cdots \tag{11}$$

Analysis of the F_d exponents reveal a repetitive and sparse structure [13], and it can be written as:

$$F_d=\begin{bmatrix}
\mathbf{I_{d_3}} & \mathbf{\Phi}_{12} & \mathbf{\Phi}_{13} & \mathbf{\Phi}_{14} & \mathbf{\Phi}_{15} & \mathbf{0}_{3\times6} \\
\mathbf{0}_{3\times3} & \mathbf{I_{d_3}} & \mathbf{\Phi}_{23} & \mathbf{\Phi}_{24} & \mathbf{\Phi}_{25} & \mathbf{0}_{3\times6} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{\Phi}_{33} & \mathbf{\Phi}_{34} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times6} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I_{d_3}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times6} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I_{d_3}} & \mathbf{0}_{3\times6} \\
\mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{I_{d_6}}
\end{bmatrix} \tag{12}$$

With Q_c being the noise covariance matrix $Q_c=diag(\sigma_{n_a}^2\cdot I,\sigma_{n_{b_a}}^2\cdot I,\sigma_{n_\omega}^2\cdot I,\sigma_{n_{b_\omega}}^2\cdot I)$, the covariance matrix Q_d can be obtained by the discretization of Q_c .

$$Q_d=\int_{\Delta t}F_d\left(\tau\right)G_c Q_c G_C^T F_d(\tau)^T d\tau \tag{13}$$

With F_d and Q_d , the covariance matrix can be computed by the filter equation

$$P_{k|k-1}=F_d P_{k-1|k-1}F_d^{\ T}+Q_d \tag{14}$$

Therefore, with the discretized error state propagation and error process noise covariance matrices, the state can be propagated as follows:

- When IMU data ω_m and a_m arrived in a certain sample frequency, the state vector is propagated using numerical integration on Equation (5).
- Calculate F_d and Q_d .
- Compute the propagated state covariance matrix according to the filter Equation (14).

III. Posture Recovery from Camera

To provide the measurement update from the camera in mobile devices. A brief outline of the processes are depicted as follows.

A. Feature detection.

During the feature-detection step, the image is search for salient key-points that are likely to match well in other images. An overview of different detectors can be found in [14]. In order to obtain the real-time performance, we chose ORB [15] for feature detection. It extremely fast to provide features with good invariance to changes in viewpoint and illumination. In order to improve the robustness of the pose calculation, we divide the image into uniform grids for feature detection.

B. Feature tracking.

Feature tracking is an extremely important for our motion tracking system. To make a tradeoff between the robustness and efficiency in tracking, we combine the detect-then-track and the KanadeLucasTomasi (KLT) [8] tracker in a flexible complementary. As the name implied, the detect-then-track method is to detect features in the first image, and then, search for their corresponding matches in the following image. With respect to the KLT algorithm, the detect-then-track can provide matching features with more accurate. However, even with the ORB feature extraction, the computation cost for image stream is still a great challenge for mobile devices. Fortunately, due to the continuous image frame, the difference between neighbor images is relative small, which creates a nice preparation for the implementation of the more efficient KLT algorithm.

In this paper, we combine the mutual advantages of ORB feature-extraction and KLT for a robust and efficient feature tracking. Given a series of image stream, the ORB feature detection is perform on certain images with great changes, while the KLT tracks the image stream with smooth. The entire feature tracking strategy is depicted in **Algorithm 1**.

Algorithm 1: Feature-based Tracking Strategy

```
Given the threshold number of the features $N_{th}$, which can be adjust with the scenes
for $k = 1,2,\cdots$ do
    if $k < 3$
        { ORB feature extraction with feature number $N$
          Feature tracking = detect-then-track}
    else
        { if $N \geq N_{th}$
              Feature tracking = KLT tracking;
          else if $N \leq N_{th}$
              Feature tracking = detect-then-track;
          end }
    end
end
```

With the feature tracking algorithm above, the vision-based tracking performance in the mobile device can acquire more efficient.

C. Posture recovery from camera.

According to the feature tracking strategy in the last section, the corresponding transformation between two continuous images can be recovered by epipolar geometry. As shown in Fig. 2, the epipolar geometry describes the relationship between two neighboring images [16].

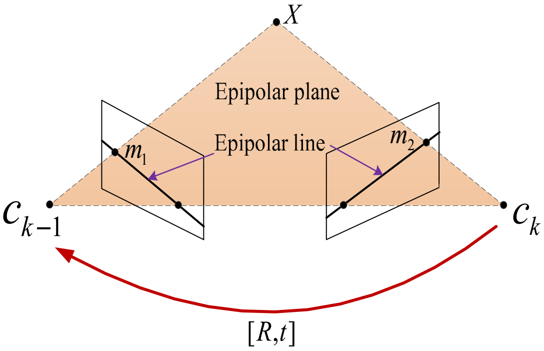


Fig. 2 The epipolar geometry between two views

According to the matching point sets (m_{k-1}, m_k) in two images, the fundamental matrix F can be derived with $m_k{}^T F m_{k-1} = 0$. The rotation and position matrix of the frame c_k with respect to c_{k-1} are denoted by R and t, respectively. The fundamental matrix F is defined as $F = K^{-T} R^T \lfloor t \rfloor_\times K^{-1}$, where K is the intrinsic parameters of the camera which can be obtained by the camera calibration method [17] in advance. Generally speaking, the matched points are usually contaminated by some outliers, and the ORB feature extraction is more inclined to noise for its efficiency. For more accurate camera motion trajectory estimation, it is important to exclude the outliers. The RANSAC [18] has been established as the standard method for the model estimation in the presence of outliers, which can remove the outlier matching points and improve the robust for fundamental matrix F calculation. With these methods, a series of the camera poses can be recovered along the trajectory when new image stream arrived.

IV. Measurement Update

A. Measurement model

With the visual algorithm in section III, the localizations of the camera trajectory can be obtained. With respect to the low cost IMUs in mobile devices, the integrated drift over time may lead the motion tracking collapsed due to the bias and noise inherent. Therefore, postures of the camera from visual algorithm are considered as the measurements in the error-state Kalman Filter framework. For the camera position measurement p_w^c, the following measurement model can be obtained

$$z_p = p_w^c = p_w^i + C_{(q_w^i)}^T p_i^c + n_p \tag{15}$$

Where, $C_{(q_w^i)}$ as the IMU's attitude in the world frame, n_p is the measurement noise of the position, and the position error can be defined as:

$$\tilde{z}_p = z_p - \hat{z}_p \tag{16}$$

The equation (16) can be linearized as follow.

$$\tilde{z}_{pl} = H_p \tilde{x} \tag{17}$$

At the same time, the orientation of camera can be derived by the error quaternion. The rotation from camera frame to world frame yielded from visual odometry is q_w^c, and we can model this as:

$$z_q = q_w^c = q_i^c \otimes q_w^i \tag{18}$$

Therefore, the error measurement can be acquired:

$$\tilde{z}_q = z_q - \hat{z}_q = z_q \otimes \hat{z}_q{}^{-1} = (q_i^c \otimes q_w^i) \otimes (q_i^c \otimes \hat{q}_w^i) \tag{19}$$

Finally, the measurements can be stacked next:

$$\begin{bmatrix} \tilde{z}_p \\ \tilde{z}_q \end{bmatrix} = \begin{bmatrix} H_p \\ H_q \end{bmatrix} \tilde{x} \tag{20}$$

B. Overall motion tracking strategy

According to above process, the measurement update can be obtained, and the well-known Kalman Filter procedure is applied to execute the sensor fusion. Finally, the entire iterative loop can be summarized as **Algorithm 2**.

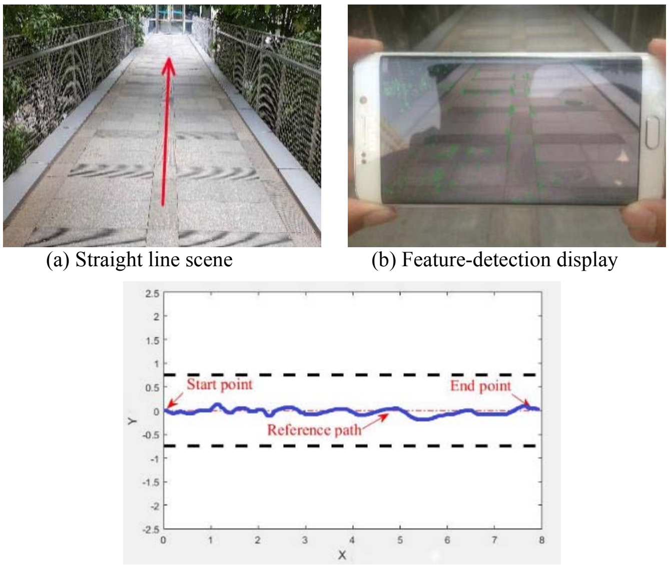
Algorithm 2: Motion Tracking Process

```
Initialize $\hat{x}_{0|0}$, $\tilde{x}_{0|0}$ and $P_{0|0}$
for $k = 1, \cdots$ do
    { Time update:
        Compute $F_d$ and $Q_d$
        $\tilde{x}_{k|k-1} = \mathbf{0}_{21\times1}$, $P_{k|k-1} = F_d P_{k-1|k-1} F_d{}^T + Q_d$
        Compute $\hat{x}_{k|k-1}$ with the 4th Runge Kutta integration }
    if Pose from visual algorithm arrived
        {Measurement update:
         Compute the residual: $\tilde{z} = z - \hat{z}$
         Compute Kalman gain: $K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1}$
         Compute the correction: $\tilde{x}_{k|k} = \tilde{x}_{k|k-1} + K_k \tilde{z}$
          $P_{k|k} = (I_d - K_k H) P_{k|k-1} (I_d - K_k H)^T + K_k R K_k{}^T$
         Use $\tilde{x}_{k|k}$ to correct state estimate and the obtain $\hat{x}_{k|k}$
    end }
end }
```

V. Experimental Results

The proposed method is evaluated by using a mobile device SAMSUNG S6 edge, and the computing power is far behind the current personal PC. Even with this typical resource constrained mobile device, our motion tracking system can run in real time (about 25 fps).

We have done two experiments on the yard outdoor. The first one is performed on a straight passage. Currently, the lack of a reliable ground truth makes only a qualitative evaluation of our motion tracking. With the smartphone hold by hands, the pedestrian walked along with the narrow center brick for a reference trajectory, as the red arrow showed in Fig. 3 (a). Fig. 3 (b) shows the feature detection and motion tracking procedure performing on the smartphone, the features are marked as green dots. The blue curve in Fig. 3 (c) show the actual motion tracking trajectory derived by our proposed method. Even with no ground truth for evaluation, we can find that the tracking trajectory of our method closed to the ideal red reference line with dash dot, which respects the reference path of the red arrow in Fig. 3 (a).

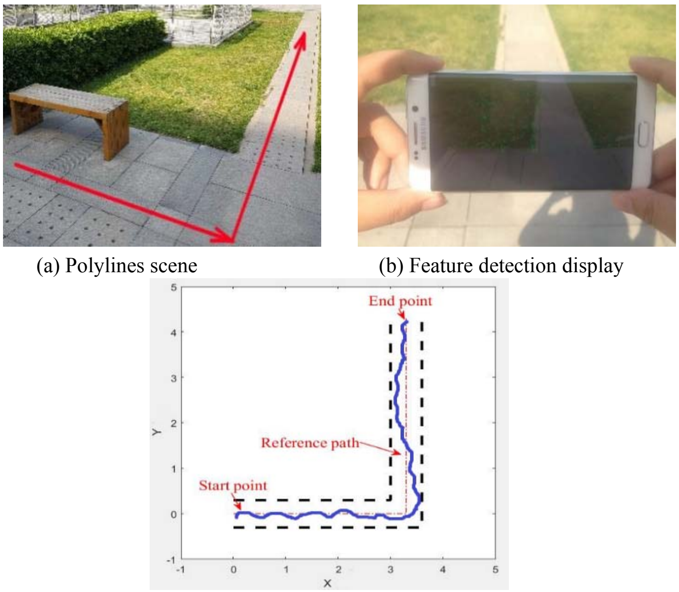


(a) Straight line scene (b) Feature-detection display

(c) Actual motion tracking path

Fig. 3. Motion tracking in straight line scene

To verify the proposed method further, the motion tracking experiment is performed along a trajectory with two lines orthogonal. The red arrow-line in Fig. 4 (a) respects the ideal reference path of the pedestrian. The actual working situation of the mobile device is displayed in Fig. 4 (b), and with the proposed motion tracking method, the predicted travel trajectory is derived as blue curve in Fig. 4 (c). We can also find that the deviation reaches maximum at the corner, where the vision-based motion tracking would collapsed due to great scene changes. However, the proposed motion tracking strategy can overcome the challenge and then convergence close to the reference path. Since the irregular shake of the hand-hold mobile device due to the pedestrian's walking, the predicted tracking path is not smooth enough, but the tendency of the actual tracking path can also demonstrated the validity of the method proposed in the paper.



(a) Polylines scene (b) Feature detection display

(c) Actual motion tracking path

Fig. 4. Motion tracking in polyline scene

VI. CONCLUSIONS

In this paper, we show a vision-aided inertial motion tracking method in real time on mobile devices. Due to the limited computing power in mobile device, the focal flow and the ORB features extraction are effectively combined together, which ensures a tradeoff between the robustness and efficiency. In addition, the error state Kalman Filter is selected for our sensor fusion due to its low computational complexity. In order to compensate the drift of IMU due to time integration, the camera embedded in the mobile device provides correction from visual algorithm. The experimental results indicate that the drift of IMU can be reduced by the vision algorithm and the predicted trajectory is close to the actual one, which also demonstrate the effectiveness and feasibility of the proposed motion tracking method. Given the proposed position tracking method, the navigation can be performed to acquire full 6 degree of freedom, which would become an indispensable component for the large-scale virtual reality application or other autonomous equipment. In the next step, we will focus on the proposed motion tracking system to support production and process planning, it is helpful to fulfill the activity recognition for human computer interaction in digital manufacturing.

REFERENCES

[1] E. Marchand, H. Uchiyama, F. Spindler, "Pose estimation for augmented reality: a hands-on survey." IEEE Transactions on Visualization and Computer Graphics, PrePrints, doi:10.1109/ TVCG2015.2513408.

[2] J. Levinson, J. Askeland, J. Becker, J. Dolson., "Towards fully autonomous driving : systemsand algorithms," 2011 IEEE Intelligent Vehicles Symposium, 2011, pp.163-168.

[3] S. Weiss, M.W. Achtelik, S. Lynen, et al, "Monocular vision for long-term micro aerial vehicle state estimation: a compendium," Journal of Field Robotics, vol. 30, no. 5, 2013, pp.803–831.

[4] F. Geiselharta, M. Ottob, E. Rukzioa, "On the use of Multi-Depth-Camera based Motion Tracking Systems in Production Planning Environments", 48th CIRP conference on manufacturing systems, 2016, pp.759-764.

[5] S. Sukkarieh, E. Nebot, H. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," IEEE

Transactions on Robotics and Automation, vol. 15, no. 3, 1999, pp. 572–578.

[6] D. Nister, O. Naroditsky, and J. Bergen, "Visual Odometry," IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 652-659.

[7] D. Scaramuzza, F. Fraundorfer, “Visual Odometry: part I - The first 30 years and fundamentals,” IEEE Robotics and Automation Magazine, vol. 18, no. 4, 2011.

[8] S. Baker, I. Matthews, “Lucas-Kanade 20 years on : a unifying frame,” International Journal of Computer Vision, 2004, pp. 221–255.

[9] M. Bosse, W. Karl, D. Castanon, P. DeBitetto, “A vision augmented navigation system,” IEEE Conference on Intelligent Transportation System, 1997, pp. 1028–1033.

[10] L. Armesto, J. Tornero, M. Vincze, “Fast ego-motion estimation with multi-rate fusion of inertial and vision,” International Journal of Robotics Research, vol. 26, no. 6, 2007, pp. 577–589.

[11] V. Madyastha, V. Ravindra, S. Mallikarjunan, A. Goyal, "Extended Kalman Filter vs. Error State Kalman Filter for aircraft attitude estimation", AIAA Guidance and Navigation Control Conference, 2011, 6615.

[12] P. Furgale, J. Rehder, R. Siegwart, “Unified temporal and spatial calibration for multi-sensor Systems,” IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp.1280-1286.

[13] S. Weiss, R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," IEEE International Conference on Robotics and Automation, 2011, pp. 4531-4537.

[14] R. Siegwart, I. Nourbakhsh, D. Scaramuzza, “Introduction to autonomous mobile robots,” 2nd ed, Cambridge, MA, MIT Press, 2011.

[15] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: An efficient alternative to SIFT or SURF", IEEE International Conference on Computer Vision, 2011, pp. 2564-2571.

[16] R. Hartley and A. Zisserman, “Multiple View Geometry in computer vision,” 2nd ed, Cambridge University Press, 2008.

[17] Z. Zhang, “A flexible new technique for camera calibration,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, 2000, pp. 1330-1334.

[18] C., Sunglok, T. Kim, and W. Yu. "Performance evaluation of RANSAC family." In Proceedings of the British Machine Vision Conference, 2009.