2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

An Efficient Pose Estimation for Limited-Resourced MAVs using Sufficient Statistics

Ilankaikone Senthooran[1,†], Jan Carlo Barca[1,†], Joarder Kamruzzaman[2,†], Manzur Murshed [2,†], Hoam Chung[3,†]

Abstract— **We present a computationally efficient RGB-D based pose estimation solution for less computationally resourced MAVs, which are ideally suited as members in a swarm. Our approach applies the sufficient statistics derived for a least-squares problem to our problem context. RANSAC-based outlier detection in aligning corresponding feature points is a time consuming operation in visual pose estimation. The additive nature of the used sufficient statistics significantly reduces the computation time of the RANSAC procedure since the pose estimation in each test loop can be computed by reusing previously computed sufficient statistics. This eliminates the need for recomputing estimates from scratch each time. A simpler hypotheses testing method gave similar performance in terms of speed but less accurate than our proposed method. We further increase the efficiency by reducing the problem size to four dimensions using attitude data from an Attitude and Heading Reference System (AHRS). Using a real-world dataset, we show that our algorithm saves up to 94% of computation time for the RANSAC-based procedure in pose estimation while improving the accuracy.**

I. Introduction

Autonomous swarms of Micro Aerial Vehicles (MAVs) have numerous indoor applications such as surveillance, monitoring, collapsed building exploration and aiding in disaster relief operations. In order to support robustness and graceful degradation of performance in the case of individual failure, a swarm robotic system comprises of individuals that are less capable and dispensable. Thus in a swarm of MAVs, individuals are very limited in resources compared to a single specialised MAV with many sensors and high computational resources.

Achieving autonomy of an MAV begins with estimating its position and orientation in the environment, known as pose estimation. The computational speed of pose estimation is more crucial in aerial robots than in ground robots since the performance in pose estimation is directly linked to the stability of a vehicle. It is obvious that the problem becomes much more challenging in limited resourced MAVs because of their low processing power and memory.

Various exteroceptive sensors such as laser range finders [1]–[3], monocular cameras [4], [5] and stereo cameras [6], [7] have been used effectively for indoor navigation in MAVs. But in recent years, RGB-D visual pose estimation methods have become popular among the MAV research community for indoor applications [8]–[11]. The main reason behind this trend is that RGB-D cameras are independent modules providing rich 3D information about its environment [12], which eliminates the need for any further processing to calculate depth information, such as triangulation in stereo cameras. In addition, their small size, light weight and economical price make them ideal for use in low-cost MAVs.

Some of the first works that used RGB-D cameras for real-time robot control did not completely rely on the RGB-D information for computing full pose estimates. For example, RGB-D information was used for estimating and controlling the flying height of an MAV [8] and for indoor exploration where pose estimation was done using a laser range finder [13]. Huang et al. [9] was the earliest notable case of using an RGB-D camera for onboard visual odometry. The more recent works [10], [11] achieved fully onboard localisation and mapping using RGB-D camera. However, all the aforementioned visual odometry methods [9]–[11] used single board computers that are relatively high in processing power and cost.

A recent comparison study [14] provided a detailed analysis and experimental comparison of several state-of-the-art real-time odometry estimation methods that use RGB-D cameras, focusing on algorithms suitable for limited-resourced MAVs. They categorised the existing visual odometry methods into three groups according to sensor data types as (1) image-based, (2) depth-based and (3) both image- and depth-based. Imaged-based methods are further subdivided as (a) sparse visual feature based methods, (b) sparse visual feature based methods combining depth data, and (c) dense feature based methods. For experimental comparison they used several existing real-time RGB-D visual odometry methods that fall into different categories mentioned above. The methods included, among others, Libviso2 [15], Fovis [9], DVO [16] and FastICP [17]. It should be noted that among the algorithms considered, only Fovis [9], which is a sparse visual feature based method combining depth data, was originally implemented on an MAV. The results from the experiments using the RGB-D dataset [18] as well as self-recorded datasets on a laptop computer show that Fovis [9] is the fastest (least CPU-intensive) method.

However, in our preliminary experiments we discovered that even a sparse visual feature based method which is very

[1] Ilankaikone Senthooran and Jan Carlo Barca are with the Faculty of Information Technology, Monash University, Clayton VIC 3800, Australia ilankaikone.senthooran,jan.barca@monash.edu

[2] Joarder Kamruzzaman and Manzur Murshed are with the School of Engineering and Information Technology, Federation University, Churchill VIC 3842, Australia {joarder.kamruzzaman, manzur.murshed}@federation.edu.au

[3] Hoam Chung is with the Department of Mechanical and Aerospace Engineering, Monash University, Clayton VIC 3800, Australia hoam.chung@monash.edu

[†] The authors are with the Monash Swarm Robotics Laboratory, Monash University, Clayton VIC 3800, Australia

978-1-4799-9993-4/15/$31.00 ©2015 IEEE

similar to Fovis [9] produces pose estimation at a rate less than 1Hz on our system. This is not fast enough for real-time navigation of our MAV swarm.

We propose an efficient pose estimation method using RGB-D and AHRS (Attitude and Heading Reference System) which enables autonomous flight of a limited-resourced MAV by achieving adequately fast estimates for real-time navigation while not compromising the accuracy. We realise this by using the sufficient statistics of least-squares minimisation to speed up the outlier detection step of the pose estimation process. These sufficient statistics has been proposed in a previous work in bio-informatics [19] for efficiently aligning protein structures. Even though pose estimation also involves a similar alignment process for computing the transformation between camera poses, the 3D feature point sets contain noisy data which are normally pruned out inside the RAndom SAmple Consensus (RANSAC) procedure. We make use of the additive nature of these sufficient statistics [19] to improve the speed of the RANSAC process by reusing previously computed transformations. To the best of our knowledge, existing works have not explored the use of sufficient statistics for pose estimation and we are the first to adopt this approach. We further improve the estimation speed by reducing the degrees of freedom (DOF) of the problem from 6 to 4 using attitude information from an AHRS.

The rest of this paper is organised as follows: Section II gives a detailed description of the proposed method. In this section we explain our motivation behind our choice to speed up the outlier detection step (II-A), how we use sufficient statistics in the RANSAC-based outlier detection (II-B) and how we incorporate AHRS information to further improve the speed of pose estimation (II-C). Experimental results are presented in Section III, which includes the system overview (III-A). Finally, concluding remarks and future work are mentioned in Section IV.

II. Pose Estimation

A. Motivation

We initially re-implemented a visual pose estimation method available in Mobile Robot Programming Toolkit[1] (MRPT) and tested it on our system (detailed in Section III-A) to investigate the pose estimation speed and time taken by each step involved. We chose to implement a visual (2D) feature based method which combines depth and visual information, since this category of methods has been identified as the fastest in [14]. A simple block diagram of this method is shown in Fig. 1.

The pose estimation process starts by prepossessing and extracting features from an RGB image using the Good Features To Track [20] method. Detected features are then tracked across successive RGB image frames using the pyramidal implementation of the Lucas-Kanade feature tracker [21]. We adopt a keyframe technique for reducing short-scale drift, where features are only detected for keyframes and then tracked across several successive frames until a new keyframe is assigned. The optical flow step yields two sets of feature points which are then projected onto 3D coordinates by using their 2D feature positions and the corresponding depth data. The transformation which projects these two sets of 3D points onto one other is related to the relative motion of the camera. However, since the two 3D point sets are most likely to contain outliers due to noise, these outliers are first removed using a RANSAC procedure. Then the transformation between these two sets of filtered 3D points is calculated and in turn used to calculate the relative pose of the camera between those frames.

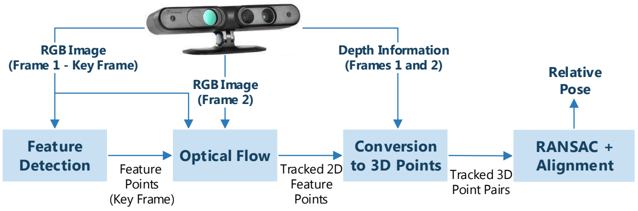

Fig. 1. Information Flow in Pose Estimation: for feature detection and optical flow commonly used OpenCV C++ implementations goodFeaturesToTrack and calcOpticalFlowPyrLK are used, respectively

TABLE I
Process execution time of a single pose estimation

Process	Feature Extraction	OpticalFlow	RANSAC+Alignment
Time	0.48s	0.06s	1.48s

Table I shows the average process execution time over 5 experimental runs conducted on our system. It is evident that the RANSAC-based outlier removal step requires the highest CPU time of all the steps, followed by the feature extraction step. The RANSAC process for which the execution time is shown in the table is explained in the following Section II-B as RANSAC-HT2. A single pose estimation cycle takes more than a second and this pose estimation rate of less than 1Hz is not fast enough for an autonomous robot. Thus we aim to increase the estimation speed by first targeting to improve on the outlier removal step which is the slowest. It should be noted that even though FAST features [22] used in Fovis [9] is faster at extracting features, a larger number of features need to be extracted to ensure that enough features are matched across frames. This in turn increases the time taken for outlier detection by a larger margin than the time saved in feature detection.

B. Applying sufficient statistics to pose estimation

RANSAC [23] is a non-deterministic algorithm for fitting a model to a dataset contaminated with outliers. It is composed of the two steps, *hypothesise* and *test* that are repeated in an iterative manner. This process includes generating several hypotheses (model parameters) using randomly selected minimal sample sets and then testing each of these hypotheses on the entire input dataset to build up consensus sets of ‘inliers’. The hypothesis with the highest score (e.g. most number of inliers) is identified as the

[1] http://www.mrpt.org/list-of-mrpt-apps/application-kinect-3d-slam/

best. Favourable properties of RANSAC include robustness, generality and simplicity. It is a widely used method for estimating geometric transformations in computer vision, and consequently used in visual odometry algorithms for approximating the transformation between feature points tracked across different image frames.

When using RANSAC for relative pose estimation with RGB-D data, minimal sample sets are randomly selected from the entire tracked 3D points, and they are used to calculate corresponding transformations (hypotheses). These transformations then need to be tested against the remaining set of tracked 3D points. One possible way to do this involves using the calculated transformation to project remaining points and computing error between corresponding point pairs [24]. Another way would be to calculate a new transformation by adding each remaining point pair to the sample set and taking the difference from the initial transformation [25]. These two methods will be hereafter referred to as HT1 and HT2, respectively. Even though this re-computation of transformations is computationally costly, our experiments showed that using HT2 for hypotheses testing has better accuracy. This motivated us to speed up RANSAC process which uses HT2 (RANSAC-HT2).

Sufficient statistics [26] are essentially a set of statistics that summarise all of the information in a sample about a certain parameter. A set of sufficient statistics with respect to the least squares alignmen was derived in [19], and furthermore, these statistics are demonstrated to be additive. Therefore, by using these statistics in the RANSAC process, computing transformations of the sample set plus each additional point pair can be performed in constant time by reusing the previous solutions of the sample set. This process is much faster than recomputing transformations from scratch. Now let us re-derive the set of sufficient statistics of [19] with respect to our problem for completeness.

Let us denote the two 3D feature sets obtained from successive RGB-D camera frames as $U = \{u_1, u_2 \ldots u_n\}$ and $V = \{v_1, v_2 \ldots v_n\}$, where n is the number of features in each set and u_i corresponds to $v_i,\ i = 1,\ldots,n$. The objective is to find the rotation R and translation t that aligns the point sets U and V while minimising the alignment error defined as

$$\min_{R,t} \sum_{i=1}^{n} \|Rv_i + t - u_i\|^2. \tag{1}$$

Since R is independent of t at the optimum, translating U and V such that their geometric centres are at the origin yields an objective function that is independent of t

$$\min_{R} \sum_{i=1}^{n} \|Rv'_i - u'_i\|^2, \tag{2}$$

where $u'_i = u_i - c(U)$ and $v'_i = v_i - c(V)$ are the corresponding translated features results in the modified feature sets U' and V'. Note that $c(U) \triangleq \sum u_i/|U|$, where $|U|$ denotes the cardinality of the set U.

Rigid-body alignment is a general regression problem and we assume that it produces error terms, $\varepsilon_i = Rv'_i - u'_i$ that are normally distributed as $\mathscr{N}(0,\sigma)$, which is minimised by the solution of (2). Thus, the likelihood of the normally distributed error terms after alignment is given as

$$f(\varepsilon_1,\ldots,\varepsilon_n|\sigma) = (2\pi\sigma^2)^{-\frac{n}{2}} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\|\varepsilon_i\|^2\right). \tag{3}$$

By examining the decomposition of $\|Rv'_i - u'_i\|^2$, the summation term in (3) can be deduced to a form containing a set of statistics which does not take into account its data explicitly. As these statistics are also sufficient to estimate σ, they form a set of sufficient statistics for the least-squares minimisation problem. This set consists of 24 distinct statistics as described below. First, let us define

$$s_m^{ij} \triangleq u'_{ij} - v'_{ij},\ s_p^{ij} = u'_{ij} + v'_{ij},\ i = 1,\ldots,n,\ j = x,y,z,$$

where u'_{ix} indicates the x component of vector u'_i and so on. From these, we can define the set of sufficient statistics as

$$\Omega = \left\{ \text{for } j,k \in \{x,y,z\}, \sum_{i=1}^{n} s_m^{ij}, \sum_{i=1}^{n} s_p^{ij}, \sum_{i=1}^{n} s_m^{ij} s_m^{ik}, \right.$$
$$\left. \sum_{i=1, j\neq k}^{n} s_m^{ij} s_p^{ik}, \sum_{i=1}^{n} s_p^{ij} s_p^{ik} \right\}. \tag{4}$$

Let us consider two pairs of corresponding feature sets, (U'_s, V'_s) and (U'_t, V'_t). Let Ω_1 and Ω_2 be the sufficient statistics of the alignment of the first and the second pair respectively. As shown through lemmas 1-3 and corollaries 1-4 in [19], sufficient statistics of least-squares minimisation are additive. Therefore, we can use Ω_1 and Ω_2 to derive set of sufficient statistics Ω' of the alignment of U' with V', where $U' = U'_s \cup U'_t$ and $V' = V'_s \cup V'_t$.

The updated sufficient statistics Ω' is now used with a method proposed in [27] to recompute rotation R. This method transforms the least-squares minimisation problem to an eigenvalue problem of the form $Q(\Omega)q = \lambda q$. $Q(\Omega)$ is a 4×4 symmetric matrix that is a function of Ω, q is the rotation R in quaternion form and λ is eigenvalue corresponding to q. The eigenvector of the minimum eigenvalue gives the optimal rotation which minimises (2). Each element of matrix $Q(\Omega)$ can be computed by combining a subset of the sufficient statistics Ω given in (4). Since these statistics are additive, matrix Q of the sample set with an additional pair of points can be constructed by finding the elements of Q for the additional pair and adding it to the previously computed Q of the sample set. This eliminates the need of recomputing the Q from entire set. Once Q is updated using sufficient statistics, it is diagonalised to find the corresponding rotation.

Thus, by adopting sufficient statistics in RANSAC-based pose estimation, the process speed can be drastically increased by performing least-square minimisation using the previous partial solutions. A single iteration of the RANSAC procedure of our proposed fast pose estimation algorithm is shown in Algorithm 1. It begins by randomly selecting a pair of sample sets (U_s, V_s) from (U, V), in which each element in U_s has a corresponding element in V_s. The remaining element pairs then constitute the test pair of sets (U_t, V_t)

where $U_t = U \setminus U_s$ and $V_t = V \setminus V_s$. The sufficient statistics in (4) are computed for the sample set. These statistics are denoted by Ω_s. Then Ω_s is used to compute the relative pose estimation p_s for the sample set by constructing the corresponding Q matrix. Next, a point pair (u_i, v_i) is picked and the sufficient statistics of that pair (Ω_i) is computed. We then compute the relative pose corresponding to the sample set with the additional point pair (p_i) from its sufficient statistics (Ω') which is computed using Ω_s and Ω_i. If the deviation of p_i from p_s is within an acceptable bound, the additional point pair is considered as an inlier and added to the consensus set (U_c, V_c). Sufficient statistics for the pair (Ω_i) is added to the sufficient statistics of the consensus set (Ω_c). At the end the iteration, precomputed values p_s, Ω_s and Ω_c are used to compute the corresponding pose estimation (p_{s+c}) for $(U_s, V_s) \cup (U_c, V_c)$.

Let us assume that we have a sample set and a test set U_s and U_t. A single iteration of RANSAC in this case consists of computing transformations for sample set plus every point pair in test set. Otherwise these transformations would have to be calculated from scratch requiring $O(|U_s||U_t|)$, whereas our method only requires $O(|U_t|)$.

Algorithm 1 Fast pose estimation with sufficient statistics

```
Input: Two sets of corresponding 3D points $U = \{u_i\}$ ,$V = \{v_i\}$, $i = 1,2 \dots n$
Output: Rotation $R$ and translation $T$.
Begin:
Randomly select minimal sample set $U_s \subset U$, $V_s \subset V$
Compute sufficient statistics $\Omega_s$ of sample set $U_s$, $V_s$
Compute relative pose $p_s$ from $U_s$ and $V_s$
for for each $u_i \in U_t = U \setminus U_s$ and $v_i \in V_t = V \setminus V_s$ do
    Compute sufficient statistics $\Omega_i$ of $u_i$,$v_i$
    Compute sufficient statistics $\Omega'$ of $(U_s,V_s) \cup (u_i,v_i)$ using $\Omega_s, \Omega_i$
    Compute relative pose $p_i$ using $\Omega'$ and $p_s$
    if $||p_s - p_i|| < threshold$ then
        add $u_i$, $v_i$ to consensus set $U_c$, $V_c$
        Update sufficient statistics $\Omega_c$ of $U_c$, $V_c$ using $\Omega_c, \Omega_i$
    end if
end for
Compute sufficient statistics $\Omega_{sc}$ of $(U_s,V_s) \cup (U_c,V_c)$ using $\Omega_s, \Omega_c$
Compute relative pose $p_{s+c}$ using $\Omega_{sc}$ and $p_s$
```

C. Further speedup of pose estimation using AHRS data

Inertial sensors are essential for MAVs and the attitude estimates from their data are accurate enough for low-level stabilisation of the vehicle. Therefore we propose the use of attitude information from an AHRS to reduce the computational load involved in pose estimation further. However, we disregard the heading information due to the presence of large errors. This is because of magnetic distortion arising from MAVs propellers and the nature of modern indoor environments [28]. By applying the attitude information from the AHRS to the alignment of 3D feature points extracted from RGB-D data, the pose estimation can be reduced to a problem with 4-DOF consisting of translation and heading. It should be noted that even though AHRS readings can be inaccurate under large angular accelerations, we assume slow and smooth motions of MAVs in this paper.

Now we describe the problem reduction steps in detail. The translated features $v'_i \in V'$ are rotated along the attitude axes (in reverse sequence: roll, pitch) by the corresponding angles provided by AHRS to get $V'' = v''_1, v''_2 \dots v''_n$. Now the alignment of the two point sets U' and V'' is essentially a least-square problem along the heading axis. To solve this in an efficient way, we use a constrained least-squares minimisation method described in [29], which we now outline.

Let $q \equiv (q_1, q_2, q_3, q_4) \equiv (q_1, \bar{q})$ be a unit quaternion that contains scalar q_1 and vector $\bar{q} \in \mathbb{R}^3$. If q is a rotation quaternion that represents a rotation along the unit vector $\mathbf{m}$ by an angle ψ, q can be represented in the following manner. $q = (\cos\frac{\psi}{2}, \mathbf{m}\sin\frac{\psi}{2}) \equiv (\cos\frac{\psi}{2}, m_x \sin\frac{\psi}{2}, m_y \sin\frac{\psi}{2}, m_z \sin\frac{\psi}{2}) \equiv (q_1, m_x q_0, m_y q_0, m_z q_0)$, where $\mathbf{m} = m_x\mathbf{i} + m_y\mathbf{j} + m_z\mathbf{k}$ and $m_x^2 + m_y^2 + m_z^2 = 1$. Then, rotating a vector v' about the rotation axis $\mathbf{m}$ by angle ψ is given by $v'' = q^{-1}v'q$. Thus, the sum of squares of the error ξ between rotated feature points in set V'' and feature points in U' can be expressed as

$$\xi = \sum_{i=1}^{n} \|u'_i - q^{-1}v'_i q\|^2. \tag{5}$$

Our objective here is to calculate rotation $q*$ that minimises the least-square error. After a series of mathematical operations, the minimisation problem can be deduced to linear equations in quaternion variables q_1 and q_0, which then can be organised into an eigenvalue problem of the form

$$\begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} q_1 \\ q_0 \end{pmatrix} = \lambda \begin{pmatrix} q_1 \\ q_0 \end{pmatrix}, \tag{6}$$

where $A = \sum_{i=n}^{n}(M_{ix}^2 + M_{iy}^2 + M_{iz}^2)$, $B = \sum_{i=n}^{n}(m_x(P_{iy}M_{iz} - M_{iy}P_{iz}) - m_y(P_{ix}M_{ix} - M_{ix}P_{iz}) + m_z(P_{ix}M_{iy} - M_{ix}P_{iy}))$, $C = \sum_{i=n}^{n}(m_x(P_{iy}M_{iz} - M_{iy}P_{iz}) - m_y(P_{ix}M_{ix} - M_{ix}P_{iz}) + m_z(P_{ix}M_{iy} - M_{ix}P_{iy}))$, $M_{ix} = u'_{ix} - v''_{ix}$, $M_{iy} = u'_{iy} - v''_{iy}$, $M_{iz} = u'_{iz} - v''_{iz}$, $P_{ix} = u'_{ix} + v''_{ix}$, $P_{iy} = u'_{iy} + v''_{iy}$, $P_{iz} = u'_{iz} + v''_{iz}$. Here, u'_{ix}, u'_{iy}, u'_{iz} are the x, y and z components of u'_i respectively. Solving the characteristic polynomial (quadratic function) of Equation 6 yields two eigenvalues and the smallest of it corresponds to $q*$ which minimises ξ. In our problem context, $\mathbf{m}$ is the unit vector along the heading axis, which is known. Thus, calculation of the heading ψ is trivial $(2\cos^{-1} q_1)$.

III. Experiment Design and Results

We implemented our pose estimation method in C++ using OpenCV[2] for feature detection (goodFeaturesToTrack) and optical flow (calcOpticalFlowPyrLK) as well as the open-source implementation[3] of [19] for alignment of 3D points and computing sufficient statistics. We evaluated the performance of the proposed method using several data sets. First, we applied our algorithm to a benchmark RGB-D data set in order to verify accuracy and computational performance on our hardware. Finally, we used our proposed method on a data set acquired through an experiment with AHRS running in real-time on our hardware. The ground truth was measured using an OptiTrack motion capture system with eight Flex 13 motion capture cameras[4] installed in an indoor environment with much texture and structure.

[2] http://opencv.org/

[3] http://www.csse.monash.edu.au/~karun/superpose3D/

[4] http://www.optitrack.com/products/flex-13/

A. System Overview

The system used in this work includes a Beaglebone Black (BBB) running Ubuntu 12.04 LTS and containing a 1GHz Sitara Cortex A8 ARM processor plus 512MB DDR3L 800MHz RAM. The RGB-D camera used is PrimeSense$^{\text{TM}}$Carmine 1.08[5], which can provide both color and depth information with a 640x480 pixels resolution at 30 frames per second. The used AHRS VectorNav$^{\text{TM}}$VN-100 Rugged[6] is capable of producing attitude data at 400Hz.

B. Test with a benchmark dataset

We evaluated the accuracy of our method on the RGB-D benchmark dataset [18], *freiburg3 long office household*. According to the tool provided with the benchmark dataset, the absolute position root-mean-square error (RMSE) of the pose estimation with sufficient statistics was 0.514m. However, as mentioned earlier, the pose estimation with RANSAC-HT1 gave higher RMSE of 0.609m. To check the accuracy of the 4-DOF pose estimation method, we assumed ground truth attitude data of the benchmark dataset as the attitude provided by the AHRS. In this case, an even better pose estimation of RMSE 0.406m was achieved as expected.

C. Computational cost comparison

We measured the computational cost of 3D feature alignment step of the pose estimation routine. Table II shows the computational cost of the alignment step for four different pose estimation routines over five test runs. These are: 1) RANSAC-HT1, 2) RANSAC-HT2, 3) RANSAC-HT2 and sufficient statistics (SS), and 4) RANSAC-HT2, SS and AHRS. It is evident that the use of sufficient statistics decreases the time required for a single feature alignment cycle, thereby decreasing overall time by 92%, leading to an average pose estimation rate of 2.31Hz on BBB and showing similar performance in terms of speed as pose estimation with RANSAC-HT1. This is three times as fast as the pose estimation without sufficient statistics (RANSAC-HT2). The use of attitude data for pose estimation decreases the time required for pose estimation further by approximately 5%.

TABLE II
EXECUTION TIME COMPARISON

Alignment	Execution Time (sec)				
	1	2	3	4	5
Standard+HT1	0.108±0.023	0.105±0.022	0.104±0.021	0.104±0.019	0.106±0.020
Standard+HT2	1.467±0.243	1.476±0.253	1.477±0.241	1.473±0.252	1.479±0.256
with SS	0.108±0.019	0.107±0.020	0.106±0.018	0.106±0.017	0.111±0.023
with SS + AHRS	0.088±0.019	0.087±0.020	0.086±0.017	0.086±0.016	0.091±0.021

D. Test with our own RGB-D and AHRS data

We conducted a series of data collection experiments with our motion capture system in place to record the ground truth. The main objective of these experiments is to validate our method using actual data from the RGB-D camera and the AHRS, since the benchmark data sets do not contain attitude information from an inertial sensor. Figure 2 shows the estimated and ground truth trajectories from one of the experiments run. The tool provided with the benchmark dataset is used to align the trajectories, upon which RMSE values of 0.130m and 0.158m are reported for the estimation with sufficient statistics and estimation using HT1, respectively. By combining the AHRS attitude data we get a better estimation with a lower RMSE of 0.095m. A detailed comparison of position and angular estimations against ground truth is given in Figure 3. For our method to be considered valid, it needs to improve on computation time while maintaining the same level of accuracy. Our method in fact even improved on the accuracy in addition to dramatically reducing computation cost.

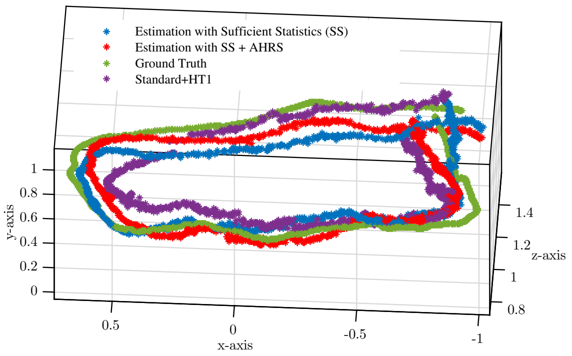

Fig. 2. Comparison of Estimated and Ground Truth Trajectories

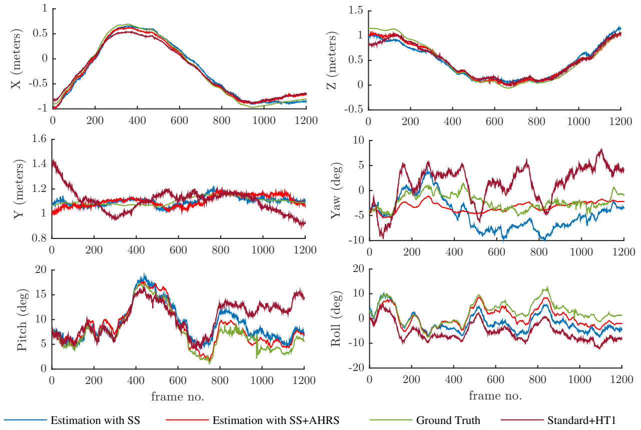

Fig. 3. Comparison of Position and Angular Estimates

IV. CONCLUSION

In this paper we present an efficient pose estimation method which is capable of producing estimates on a limited-resourced platform at an adequate rate for real-time navigation of a slowly moving MAV. This is achieved by speeding up the RANSAC-based outlier detection process by incorporating sufficient statistics to recompute pose estimations reusing previously computed information. We used two different hypotheses testing methods in the RANSAC procedure explained as HT1 and HT2. Without the use of

[5] http://www.i3du.gr/pdf/primesense.pdf
[6] http://www.vectornav.com/products/vn100-rugged

sufficient statistics, RANSAC-HT2 is slower, whereas the faster RANSAC-HT1 is less accurate. Our proposed method improved the speed of RANSAC-HT2 to a similar level as RANSAC-HT1. Pose estimation effort is further reduced by making use of attitude data from AHRS with RANSAC-HT2, bringing down the problem to only a translation and heading estimation. The accuracy of the proposed method is evaluated on a public real-world benchmark dataset and its performance has been validated by running the estimation algorithm real-time on a limited-resourced platform. Using our proposed pose estimation algorithm, we were able to reduce the time taken for the outlier detection by 94%, achieving a threefold increase in pose estimation speed compared to RANSAC-HT2 without sufficient statistics. This is even faster than RANSAC-HT1 and more accurate.

However, the absence of loop closure in the pose estimation routine makes the estimation error to grow over time. We intend to pursue implementing loop closure, integrating control and path planning as our future work to enable indoor autonomous flight with very limited resources.

ACKNOWLEDGEMENT

We are grateful to Arun S. Konagurthu of Faculty of Information Technology, Monash University for his time and effort spent in explaining to us the details of his work on orthogonal superpositions of vector sets. We also thank Madhubhashi Senanayake for all her help.

REFERENCES

[1] A. Bachrach, S. Prentice, R. He, and N. Roy, "Range–robust autonomous navigation in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[2] G. Chowdhary, D. M. Sobers, C. Pravitra, C. Christmann, A. Wu, H. Hashimoto, C. Ong, R. Kalghatgi, and E. N. Johnson, "Self-contained autonomous indoor flight with ranging sensor navigation," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1843–1854, 2012.

[3] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 1–8.

[4] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 957–964.

[5] C. Wang, T. Wang, J. Liang, Y. Chen, and Y. Wu, "Monocular vision and imu based navigation for a small unmanned helicopter," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, July 2012, pp. 1694–1699.

[6] L. García Carrillo, A. Dzul López, R. Lozano, and C. Pégard, "Combining stereo vision and inertial navigation system for a quadrotor uav," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 373–387, 2012.

[7] R. Voigt, J. Nikolic, C. Hurzeler, S. Weiss, L. Kneip, and R. Siegwart, "Robust embedded egomotion estimation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 2694–2699.

[8] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor," in *Mechatronics (ICM), 2011 IEEE International Conference on*, April 2011, pp. 358–362.

[9] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *International Symposium on Robotics Research (ISRR)*, Flagstaff, Arizona, USA, Aug 2011.

[10] S. Scherer and A. Zell, "Efficient onbard rgbd-slam for autonomous mavs," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 1062–1068.

[11] R. G. Valenti, I. Dryanovski, C. Jaramillo, D. Perea Strom, and J. Xiao, "Autonomous quadrotor flight using onboard rgb-d visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 5233–5238.

[12] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[13] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 20–25.

[14] Z. Fang and S. Scherer, "Experimental study of odometry estimation methods using rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014.

[15] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 963–968.

[16] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 3748–3754.

[17] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[19] A. Konagurthu, P. Kasarapu, L. Allison, J. Collier, and A. Lesk, "On sufficient statistics of least-squares superposition of vector sets," in *Research in Computational Molecular Biology*, ser. Lecture Notes in Computer Science, R. Sharan, Ed. Springer International Publishing, 2014, vol. 8394, pp. 144–159.

[20] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, Jun 1994, pp. 593–600.

[21] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," Intel Corporation, Microprocessor Research Labs, Tech. Rep., 2000.

[22] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, 2006, vol. 3951, pp. 430–443.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[24] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the rgb-d slam system," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 1691–1696.

[25] D. Li, Q. Li, N. Cheng, Q. Wu, J. Song, and L. Tang, "Combined rgbd-inertial based state estimation for mav in gps-denied indoor environments," in *Control Conference (ASCC), 2013 9th Asian*, June 2013, pp. 1–8.

[26] R. V. Hogg and A. Craig, *Introduction to Mathematical Statistics*, 5th ed. Prentice Hall, Dec 1994.

[27] S. K. Kearsley, "On the orthogonal transformation used for structural comparisons," *Acta Crystallographica Section A*, vol. 45, no. 2, pp. 208–210, Feb 1989.

[28] J. Qian, J. Ma, R. Ying, P. Liu, and L. Pei, "An improved indoor localization method using smartphone inertial sensors," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, Oct 2013, pp. 1–7.

[29] A. S. Konagurthu, A. M. Lesk, and L. Allison, "Minimum message length inference of secondary structure from protein coordinate data," *Bioinformatics*, vol. 28, no. 12, pp. i97–i105, 2012.