Navigation and Control for Micro Aerial Vehicles in GPS-Denied Environments

Rudolph Molero, Sebastian Scherer, Lyle Chamberlain, and Sanjiv Singh

June 20, 2011

CMU-RI-TR-10-08

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Micro-air vehicles have been increasingly employed in diverse research projects in both military and civilian applications. That is because their high maneuverability and accurate mobility. Many of them have been successfully used in outdoor areas, while some have been operated indoors. However, very few have dedicated especial attention to the case of high pitch and roll movements while doing scan-line based odometry. In this paper, we present a general approach consisting of algorithms that enable small aerial robots to fly indoors. We solve the overall problem of large movement change in pitch and roll angles by improving the standard scan matching algorithm. We also validate the effectiveness of the upgraded algorithm by a set of experiments that demonstrate the ability of a small quad-rotor to autonomously operate in cluttered indoor scenarios.

Contents

1 Introduction

Recently, the development of autonomous air vehicles has been largely incremented due to the growth of both civil and military interests. Low-cost and small-size flying machines are becoming broadly available and some of these platforms are able to lift relatively high payloads. They can be successfully utilized in several fields such us: surveillance, rescue, and research. Although great progress has been made in both control and perception for micro-air vehicles performing indoors, little has been made to address the problem of large roll and pitch maneuvers for scan-line based ladar localization.

As for autonomous ground vehicles, the main objective for autonomous flying machines is to reach a desired location in an unsupervised manner and without human intrusion [1]. Unfortunately, most indoor environments, where our research in focused on, and many other urban areas remain without access to external positioning systems, e.g. the GPS [2]. To enable indoor flight, however, we need reliable position and velocity measurements (not directly measurable), i.e. x, y, and heading. Therefore, we count on one alternative that can accurately supersede external positioning systems, i.e. the laser scanner with mathematical filters to procure position information. In addition to this, we use an Inertial Measurement Unit (IMU) which measures the rotational velocities (pitch, roll, and yaw) and magnetometer which gives us a rough idea of heading, but failures when metallic objects are close.

In order to safely maneuver flying robots in indoor environments, we need to face some critical challenges. Traditionally, unmanned vehicles operating in GPS-denied environments could rely on dead reckoning for localization, but these measurements cause a drift problem over time [3]. Another problem that may occur while flying is the existence of large pitch and roll movements yielded by dynamic instability, time-changing parameters, or wind disturbances, which can seriously undermine the pose estimation accuracy [4]. While ground vehicles do not have payload constraints, small air vehicles have significant payload limitations, which leads us to obtain the fastest algorithm possible in a computationally limited processor.

Therefore, we propose an algorithm that boosts the classic scan matching algorithm by optimizing inner processes and incorporating scan rejection criteria. This algorithm is based on the so-called *Iterative Closest Point (ICP)* algorithm. The improved technique addresses the problem of drifting, prevents the incorporation of counterfeit information during large roll and pitch angles, and maximizes the computational processing velocity.

The reminder of the paper is organized as follows. In Section II, we describe an overview of related publications. In Section III, we describe the problems of indoor flight. Then, the standard scan matching algorithm, its enhancements, and modifications are presented in Section IV. Finally, in section V, we conclude with demonstrative experiments that support our approach.

2 Related Work

Current techniques to enable micro-air vehicles frequently focus on solving the following problems: limited sensing payload, indirect odometry at high accuracy and at high frequency, limited computation on-board, fast dynamics, need to estimate velocity, and obstacle detection [5]. Initially, the emphasis was on obtaining a model and therefore an appropriate controller for this model. For example, Altug et al. [4] proposes a 3D model and two control strategies: feedback linearization and backstepping control law, but the last one proved better performance. The helicopter could not be fully autonomous, because it was restricted to vertical and yaw motions; however, one important achievement was that it employed a ground camera to estimate pose (position and orientation). Later, Roberts et al. [6] was not only able to achieve both stability and altitude control, but also collision avoidance and anti-drift control. What is more, he contributed with an approach for automatic landing and accomplished autonomous operation in indoor environments. Similarly, Bouabdallah et al. [7] not only developed innovative control techniques to redesign the flying vehicle with optimized mechanics, but also included a sonar-based obstacle avoidance and a vision-based position controller. In addition, he tested his refinements to determine that coaxial configuration was the best configuration over the quadrotor configuration. In our case, we use a quadrotor with an internal attitude and altitude controller.

Subsequent developments have focused on state estimation and Simultaneous Localization and Mapping (SLAM) in GPS-denied scenarios. For instance, He et al. [2] implemented a motion planning algorithm for a quadrotor without a GPS. It utilized Belief Roadmap (BRM) algorithms to plan trajectories. The quadrotor uses an existing map to localize and laser range data. Bachrach et at. [3], however, achieved autonomous exploration and mapped unstructured and unknown environments without requiring a prior map. Likewise, Achtelik et al. [5] achieved autonomous exploration and mapping; furthermore, he was able to locate objects of interest in unknown indoor environments. Grzonska et al. [1] proposed a similar approach to navigate indoors (localization, altitude, and attitude estimation), but he included a deflecting mirror to deviate the laser beams so as to determine the distance from the ground.

While the approaches described above [5], [1], [2] use *gmapping* algorithms for ground robots to estimate pose, our approach is designed for flying vehicles, because we consider the high roll and pitch angle movement as well as fast heading dynamics that are not significantly present in ground vehicles.

In contrast to these approaches, our system addresses the problem of large pitch an roll flight which is common in real executions. Additionally, our approach made pose estimation more robust while landing and take-off in order to avoid initial or final disorientations.

3 Indoor Flight Challenges

Like in outdoor flight, the main objective of indoor flight is to autonomously reach and maintain a desired position in a certain environment. In order to meet this goal in GPS-denied environments, we need to modify the existing algorithms for ground vehicles according to new restrictions and problems aroused in indoor flight. One main problem is pose estimation, in our case, X, Y, and heading. For this, we rely on an IMU and laser scanner, which are located on-board. The inner IMU and the embedded controller govern pitch and roll angles. The laser scanner data, on the other hand, is used to obtain the pose by employing the so-called *Scan Matching (SM)* algorithm.

Scan Matching could be done to do both mapping and control. While Scan Matching for mapping can be carried out slowly in order to gather as much information as possible so as to compute a more accurate map, Scan Matching for control has to be rapidly executed so as to provide sufficiently fast updates to the controller. For most of the applications, it is sufficient to perform Scan Matching for mapping off-line on a recorded sequence of measurements, but SM for control has to be necessarily done on-line.

Since the Standard Scan Matching algorithm is computationally slow and vulnerable to counterfeit input information, it was essential to optimize the mentioned algorithm by means of inner enhancements as well as external modifications. The misleading input information is yielded when the laser scanner suddenly points at a direction that is relatively deviated (large pitch or roll angles) from the horizontal area of work as shown on Figure 3.0.1. When these two sets of data are matched, the overall state estimations can be harmfuly affected. That is why, our algorithm has an attitude-based scan rejection module to significantly solve the problem of SM correspondence when large roll and pitch angles occur and when quick heading change occurs.

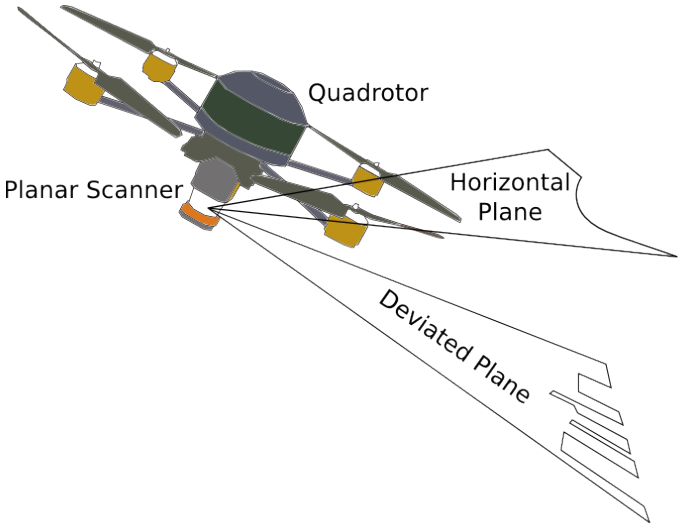

Figure 3.0.1: Problem of 2D Scanners on an aerial Vehicle. It can be clearly seen that the data from horizontal plane cannot be matched with the data from the deviated plane. Therefore this deviated data has to be rejected.

4 Approach

In this section, we present the standard Scan Matching algorithm, commonly defined by the ICP algorithm. Then we explain the enhancements that make the ICP algorithm faster and more robust. Finally, we present some modifications to optimize the algorithm within the flight restrains.

4.1 The ICP algorithm

The Iterative Closest Point algorithm is intended to align two partially overlapping clouds of points based on random guesses of the relative transformation between the clouds. The groups of points are identified as scene and model. The first one represents the set of points to be aligned, whereas the second set represents the reference to which the scene has to be aligned. After several iterations (rotations and translations) the ICP algorithm is capable to overlap two clouds of points of the same sort, that is they both belong to the same 2D scanned region as shown in Figure 4.1.1.

The ICP algorithm establishes correspondences between points on the scene and model. One way to determine corresponding points is by finding peers which have the closest distance of separation as shown in Figure 4.1.2. Later, we will use these correlations to guess transformation relations and determine the matching error.

Lara et al. [8] shows the ICP algorithm that is widely used for 2D and 3D scan matching. We have modified this algorithm by a series of enhancements and modifications as shown in Figure 4.1.3. For example, when the scene set is admitted, this has to be approved by the Scan Rejection module in order to become the next model set. The rest of enhancements and modifications will be explained in the next section.

4.2 Algorithm Enhancements

4.2.1 Neighborhood Search

In order to find correspondences between two set of points, we establish closest point relationships. One way to find the closest point for a certain point in a set is by minimizing the distance between this point in the reference-scan to all the points in the scene-scan. However, this calculus has a high computation cost, because of its O(N^2) complexity, quadratic with the shape

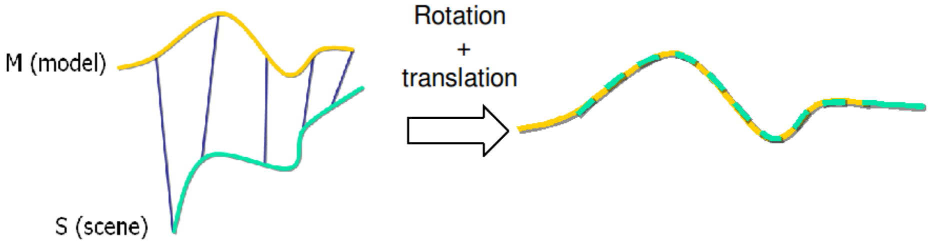

Figure 4.1.1: The ICP algorithm alighns two corresponding clouds of points (Model and Scene) by finding and applying a certain Rotation and Translation.

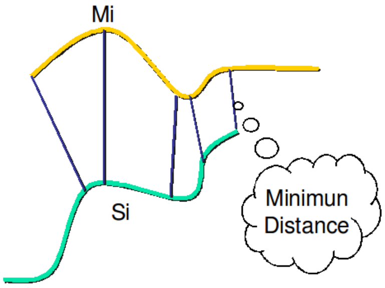

Figure 4.1.2: One way to find correspondences between the elements from Model and Scene is by the Closest point criterion. These corresponding set of points will be used to estimate rotations and translations and to calculate alighnment errors.

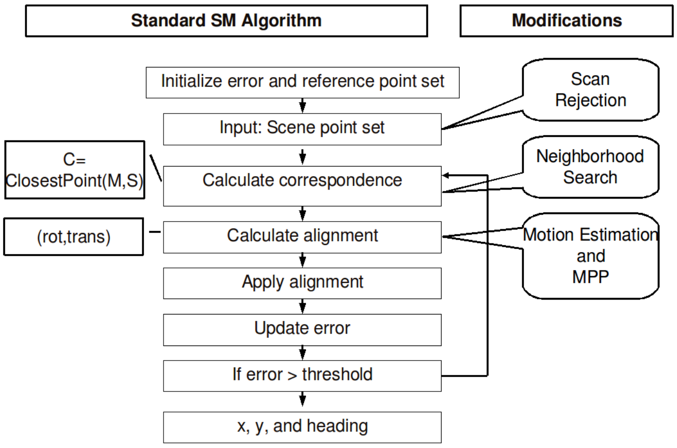

Figure 4.1.3: Standard scan matching (SM) flowchart and contributions. MPP stands for most probable pose.

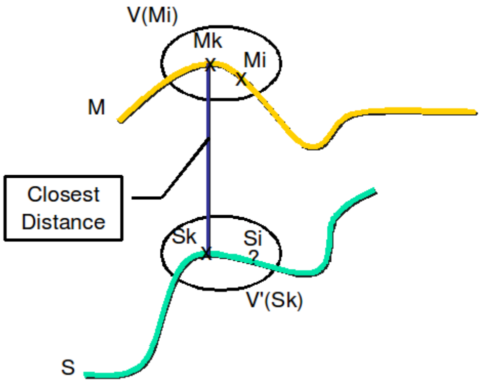

Figure 4.2.1: To find the closest distances between points from Model and Scene, we do not need to use the entire point clouds. We can do it by establishing Neighborhood search areas. For this, we first require a *correct* correspondence (M_k,S_k). Then if we need to find the closest point of M_i (provided that M_k is in its vicinity), we only need to look for its closest point in the Vicinity of S_k ($V'(S_k)$).

size N. Therefore, it is essential to reduce this processing time. Jost et al. [9] reviews several of the existing methods and proposes a new less complex ICP algorithm that uses a heuristic approach to find the closest point. Figure 4.2.1 shows the general view of the neighborhood search technique, which can be summarized as follows:

- Inputs: data sets M and S with associated neighborhoods: V(M) and V'(S).
- Outputs: for each $\mathrm{M_i}$ of M, there is an approximation $\mathrm{S_i}$ of its closest point in S.
- Procedure: $\mathrm{M_k}$ is the closest point of $\mathrm{S_k}$. If $\mathrm{M_k}$ belongs to $\mathrm{V(M_i)}$, then $\mathrm{S_i}$, the closest point of $\mathrm{M_i}$, belongs to $\mathrm{V'(S_k)}$.

It is important to mention that the set of correspondences is used to both estimate transformation matrices and calculate the error. By randomly choosing three points in the reference-scan and new-scan, it is possible to calculate an approximate transformation matrix that allows to extract the change in x, y, and heading. In addition, based on the set of correspondences, we can estimate the alignment error by summing the distances between all the corresponding points.

4.2.2 Motion Estimation

Pose stabilization does not require the knowledge of the absolute location of the vehicle in the environment [1]. As a consequence, we can estimate the relative pose change between two subsequent scans. We use the current estimated pose that is calculated by means of scan matching procedure to predict x, y, and heading change. Figure 4.2.2 shows an overview of the pose prediction algorithm. We implemented a three-step Kalman predictor proposed by Karlsson et al. [10]. This predicted pose gives an initial guess for the transformation matrix. Besides, this matrix permits to roughly align the new-scan to the reference-scan, then an initial error can be computed between the two aligned scans. Subsequent error calculations will converge based on the initial computed error.

4.2.3 Most Probable Angle Technique

After the first pose is estimated by the Kalman predictor, following transformation matrices are randomly found. We calculated a number possible changes in pose (x, y, and heading) as well

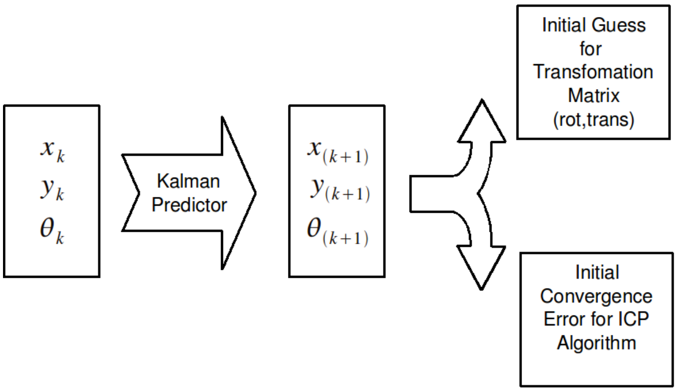
x_k
y_k
θ_k
Kalman Predictor
$x_{(k+1)}$
$y_{(k+1)}$
$\theta_{(k+1)}$
Initial Guess for Transfomation Matrix (rot,trans)
Initial Convergence Error for ICP Algorithm

Figure 4.2.2: Kalman filter is set up to estimate x, y, and heading. It permits to have two things: an initial guess for the transformation matrix and an initial convergence error for the ICP algorithm.

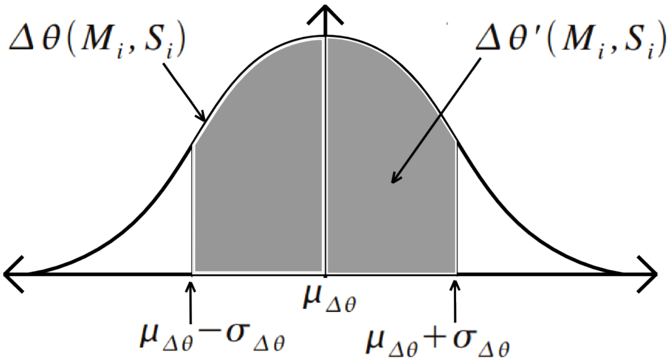
$\Delta\theta(M_i, S_i)$
$\Delta\theta'(M_i, S_i)$
$\mu_{\Delta\theta}$
$\mu_{\Delta\theta} - \sigma_{\Delta\theta}$
$\mu_{\Delta\theta} + \sigma_{\Delta\theta}$

Figure 4.2.3: Random theta change estimations can approximately be visualized in the Gaussian bell, because they have a normal distribution.

as their standard deviation and media. Although, all these calculations represented the same pose change, we discarded the cases that are far from the media by the following expressions:

$$\mu_{\Delta\theta} - \sigma_{\Delta\theta} < \Delta\theta(M_i, S_i) < \mu_{\Delta\theta} + \sigma_{\Delta\theta}, \tag{4.2.1}$$

$$\mu_{\Delta x} - \sigma_{\Delta x} < \Delta x(M_i, S_i) < \mu_{\Delta x} + \sigma_{\Delta x}, \tag{4.2.2}$$

$$\mu_{\Delta y} - \sigma_{\Delta y} < \Delta y(M_i, S_i) < \mu_{\Delta y} + \sigma_{\Delta y}. \tag{4.2.3}$$

The most probable pose (x, y, and heading) can be approximated by the pose that minimizes the alignment error. Another way to show the rejection of pose change estimations is shown in Figure 4.2.3 (only the case for heading is depicted), where $\Delta\theta(M_i, S_i)$ can be approximately accommodated in the Gaussian bell and $\Delta\theta'(M_i, S_i)$ is the result of applying the inequality 4.3.1.

4.3 Modifications

4.3.1 Attitude-Based Scan Rejection

During real excursions with the quadrotor, large pitch and roll angles take place because of wind disturbance or sudden control command changes. When it happens, the laser sensor which was

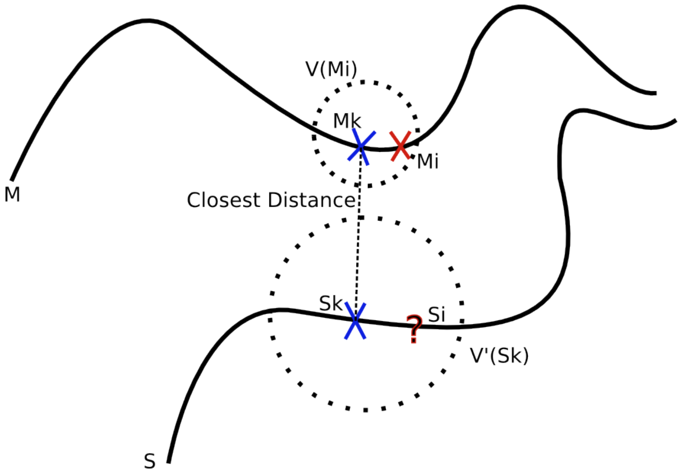

Figure 4.3.1: In order to increase accuracy of calculated closest distances, we increment the vicinity of Sk (V'(Sk)) while reducing the Vicinity of Mi (V(Mi)). This is because Mi is definite(set up by the algorithm) while Si is not indefinite (obtained by the Closest-distance algorithm).

pointing at a horizontal plane deviates its work plane in such a way that it angles the former work plane (see Figure 3.0.1). One major problem that large roll and pitch causes is that the ladar might obtain information that cannot be aligned to a previous reference. Another problem is that this bad alignment can yield error propagation, and consequently wrong pose estimation. In that sense, it is necessary to both reject those data sets and ignore them as reference-scans. The scan rejection module was based on the following condition:

```
if (pitch>p_threshold || roll>r_threshold)
        Scene Scan is Rejected
endif
```

4.3.2 Optimizations

The optimizations were made by a careful analysis of the simulation results. They enhanced our algorithm performance.

Every new scan is processed according to a certain model scan to obtain the transformation matrix. We considered the model update every ten scans so as to maintain the accuracy and speed of the algorithm. When the model needed to be updated, but the new scan used to update the model was rejected, then the former reference scan was kept until the new scan was not rejected by the Attitude-Based Scan Rejection module.

In the Neighborhood Search module, as we can see in Figure 4.3.1, we considered a relatively small search area ($V(M_i)$) in the model set; but a large research area in the scene set. Since we were looking for the closest-distance correlation for each point in the vicinity of M_i, by expanding the search area in vicinity of S_k, we could ensure that the closest distance S_i of M_i is the most accurate.

In the Most Probable Angle module, we used a hundred guesses of x, y, and heading change. Each guess is obtained by randomly selecting three points in the model and their correspondences in the scene (see Figure 4.3.2). They are later computed to find the transformation relationship according to the following expression:

$$[Model] = \begin{bmatrix} Rot & Trans \\ 0 & 1 \end{bmatrix} \times [Scene] \tag{4.3.1}$$

The guesses will be used to obtain the Gaussian bell shown in Figure 4.2.3.

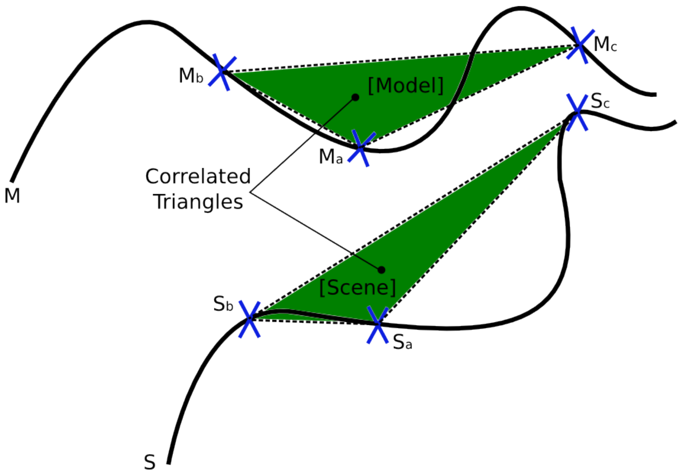

Figure 4.3.2: By randomly choosing correlating triangles and according to equation 4.3.1, we can estimate the rotations and translations that align the Scene to the Model. Then the alignment error is calculated. Afterwards, the process is repeated for about a hundred times to obtain the best alignment transformation matrix.

In the Attitude-Based Scan Rejection module, we observed that for roll or pitch angles greater than 4 deg, the alignment results were significantly deficient, and in most cases misleading; therefore, they were neglected. However, we considered these deviated scans to obtain a pose estimation only for displaying reasons.

5 Experiments

In this section, we present our quadrotor system, the comparison between our approach and the Standard Scan Matching, and finally the indoor control experiment.

5.1 Quadrotor System

Our testbed is a quadrotor helicopter which was equipped with a Linux-based embedded computer, an inertial measurement unit, and a ladar scanner (see Figure 5.1.1). The system is capable of carrying a payload of 0.4 kg. It also incorporated a flight control system for the fuselage's attitude.

During the comparing and control tests, the autopilot manually maintained a constant altitude. The Area of work comprised a cluttered environment of 20×15 m as shown in Figure 5.1.2.

5.2 Comparing Standard and Improved Scan Matching

In this section, we compare the performance of the improved Scan Matching algorithm versus the standard algorithm. The mission in both cases was to make three trips keeping the shape of a square. However, after the second square, the quadrotor is located over a reference position (starting point) that is manually reached again (final point) after making the third square. The data was collected on-board and processed off-line later on.

Since the Standard Scan Matching algorithm does not have the Scan Rejection module, it incorporates misleading information that cannot be matched. As a consequence, the resulting path tends to accumulate error in x, y, and heading. As we can see in Figure 5.2.1 (red line indicates x-direction and green line indicates y-direction), while there's no heading change, the Standard and Improved Scan Matching algorithms obtain similar results ((a) and (b)); however, when heading change and large pitch and roll movement occur, the Improved algorithm (d) significantly differs from the Standard algorithm (c). While in the improved algorithm the quadrotor points out to the right (red line), in the Standard SM, the quadrotor points out down. The heading error is approximately 70 deg, which shows that the standard algorithm notably failures, thus making 2D control unfeasible.

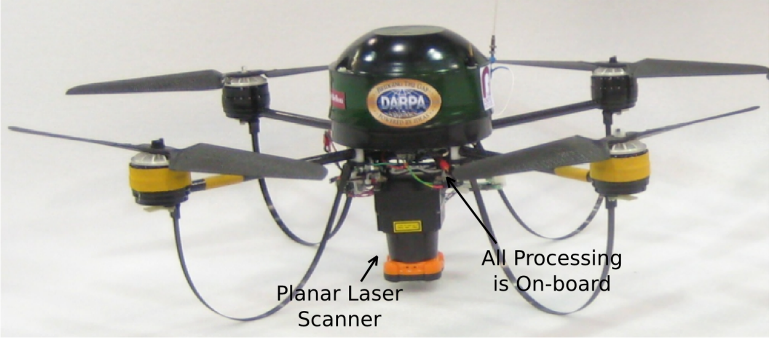

Figure 5.1.1: Quadrotor system.

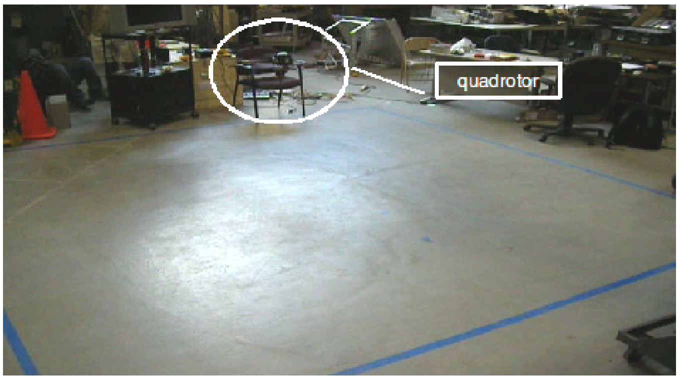

Figure 5.1.2: Data collecting environment for Figure 5.2.1. The quadrotor was operated manually.

Table 5.1: Time consumption comparison between the standard SM and the improved SM.

Technique	Time for processed scan
Standard SM	800 ms
Improved SM	60 ms

Additionally, the improved Scan Matching algorithm significantly decreased the computation time as shown in Table 5.1.

5.3 Indoor Control tests

In the experiment, we evaluated the performance of the 2D position controller. For this purpose, we first set up a desired point in a certain reference plane. Then we manually flew the quadrotor to another position. Next, we switched on the Scan-Matching based controller and the helicopter was able to autonomously return and maintain its initial set point. We successfully repeated the process three times. Figure 5.3.1 shows a snapshot of the hover control with manual perturbations. The image has four frames in which the traveled path is shown according to its condition: manual or autonomous, red and green parts respectively. It can also be seen how the quadrotor returns to its set point whenever it is manually disturbed.

In Figure 5.3.2, the top part depicts x and y behavior with respect to time. The lines in magenta contains the period when the quadrotor runs autonomously. The bottom part shows x and y commands over time. Both signals were limited according to the maximum values admitted by the quadrotor's inner controller (±0.5). Additionally, x_{trim} was also set up to compensate the disbalance in payload yielded by the weight of the laser sensor. The execution video of the algorithm is available at:

http://www.youtube.com/watch?v=itFSGZGaSu0

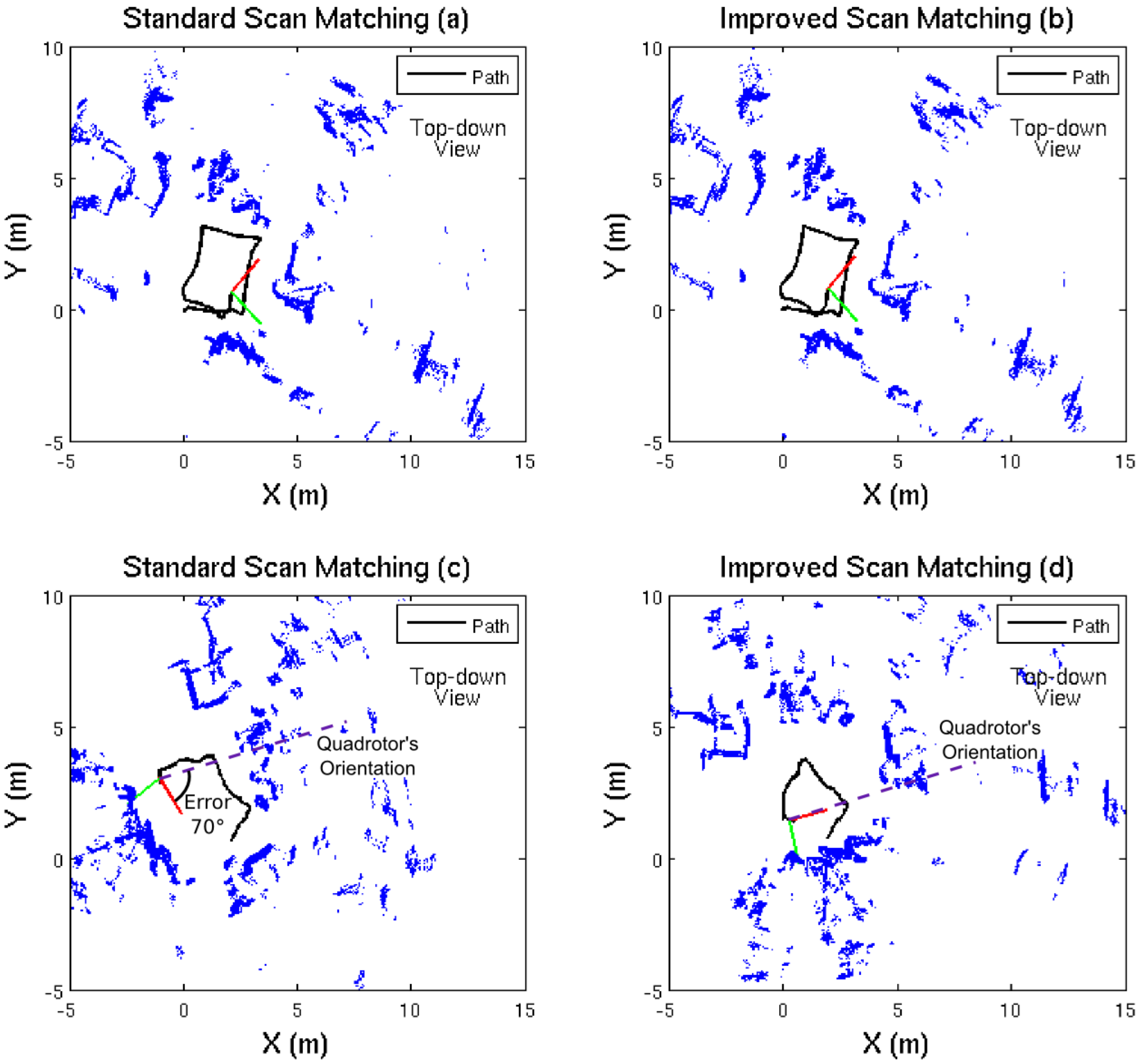

Figure 5.2.1: Comparison of the Standard SM versus our improved approach. (a) Standard SM with not heading change. (b) Improved SM with no heading change. (c) Standard SM with heading change. (d) Improved SM with heading change. Blue points represent the ladar information.

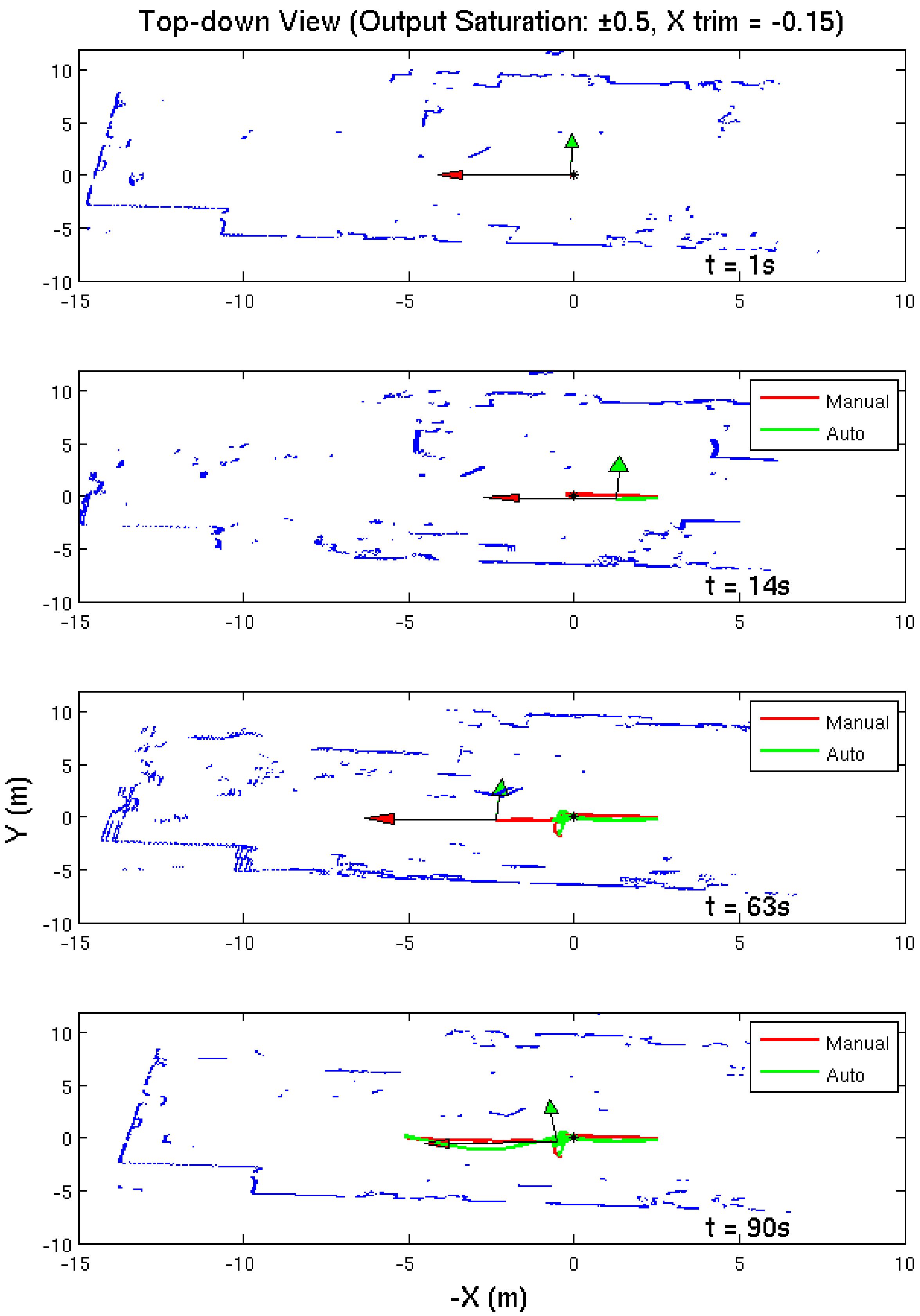

Figure 5.3.1: Hover control with manual perturbations for the quadrotor. The black mark in (0,0) indicates the desired position. The red part of the path indicates manual execution, while the green part indicates autonomous execution. Whenever the quadrotor is disturbed to a certain position, it autonomously returns to the desired position.

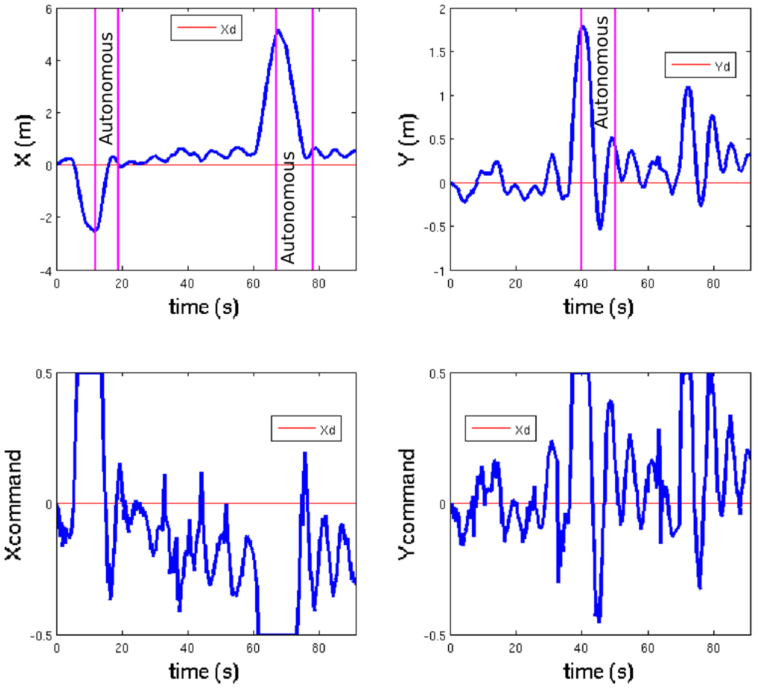

Figure 5.3.2: On the top part, we show positions X and Y over time. On the bottom, we show the commands along X and Y directions.

6 Conclusions

In this paper, we have developed a set of improvements for the standard scan matching that enables on-board position establishment. Our system includes the attitude-based scan rejection which prevents the incorporation of wrong data in the pose estimate. Moreover, neighborhood search notably reduced the computation time, because the search vicinity was significantly reduced. The Motion Estimation module that employs a Kalman filter also made the algorithm more robust, because it gave the first transformation guess and efficiently limited the convergence error. We demonstrated that the Standard Scan Matching was unable to correctly estimate x, y, and heading. Our system, in change, was capable of managing large roll and pitch angles while estimating an accurate pose. Finally, our methodology joined the set of techniques mentioned above to successfully control on-board 2D-position in unknown and unstructured GPS-denied scenarios.

We plan to do several improvements in future work. Our current method worked on a basic PD controller, therefore our next goal is to employ an H-infinity controller. This will help decrease the errors in pose estimation. Another goal is to establish a Dynamic Reference Selection. Currently, we are updating the scan reference every ten scans provided that this reference is not rejected by the Attitude-Based Scan Rejection module. However, if we dynamically select a reference scan that contains more relevant and accurate information, we will be able to improve x, y, and heading estimate. Finally, we are intending to develop algorithms that allow the quadrotor to fly interchangeably from GPS to GPS-denied scenarios. By strategically combining information between the GPS and ladar-based position estimation, we'll be able to fly the quadrotor in both indoor and outdoor environments simultaneously.

Bibliography

[1] S. Grzonka, G. Grisetti, and W. Burgard, “Towards a navigation system for autonomous indoor flying,” *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, May 2009.

[2] R. He, S. Prentice, and N. Roy, “Planning in information space for a quadrotor helicopter in a gps-denied environment,” *In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2008.

[3] A. Bachrach, R. He, and N. Roy, “Autonomous flight in unstructured and unknown indoor environments,” *International Journal of Micro Air Vehicles*, pp. 217–228, 2009.

[4] E. Altug, J. P. Ostrowski, and R. Mahony, “Control of a quadrotor helicopter using visual feedback,” *In Proc. of the IEEE Conf. on Robotics & Automation (ICRA)*, May 2002.

[5] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, “Autonomous navigation and exploration of a quadrotor helicopter in gps-denied indoor environments,” *Association for Unmanned Vehicle Systems International, IARC*, July 2009.

[6] J. F. Roberts, T. S. Stirling, J.-C. Zufferey, and D. Floreano, “Quadrotor using minimal sensing for autonomous indoor flight,” *European Micro Air Vehicle Conference and Flight Competition (EMAV)*, July 2007.

[7] S. Bouabdallah, M. Becker, and R. Siegwart, “Autonomous miniature flying robots: Coming soon!” *IEEE Robotics & Automation Magazine*, September 2007.

[8] C. Lara, L. Romero, and F. Calderón, “A robust iterative closest point algorithm with augmented features,” *Springer-Verlag Berlin Heidelberg*, vol. 5317, pp. 605–614, October 2008.

[9] T. Jost and H. Hügli, “Fast icp algorithms for shape registration,” *Springer-Verlag Berlin Heidelberg*, vol. 2449, pp. 91–99, January 2002.

[10] E. Karlsson, “The kalman predictor,” February 2009.