Cooperative Relative Navigation for Space Rendezvous and Proximity Operations Using Controlled Active Vision

Guangcong Zhang
School of Electrical & Computer Engineering
Institute for Robotics & Intelligent Machines
Georgia Institute of Technology
Atlanta, GA 303 32, USA
zhanggc@gatech.edu

Michail Kontitsis
School of Aerospace Engineering
Institute for Robotics & Intelligent Machines
Georgia Institute of Technology
Atlanta, GA 30332, USA
kontitsis@gatech.edu

Nuno Filipe
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
nrsf3@gatech.edu

Panagiotis Tsiotras
School of Aerospace Engineering
Institute for Robotics & Intelligent Machines
Georgia Institute of Technology
Atlanta, GA 30332, USA
tsiotras@gatech.edu

Patricio A. Vela
School of Electrical & Computer Engineering
Institute for Robotics & Intelligent Machines
Georgia Institute of Technology
Atlanta, GA 30332, USA
pvela@gatech.edu

Abstract

This work aims to solve the problem of relative navigation for space rendezvous and proximity operations using a monocular camera in a numerically efficient manner. It is assumed that the target spacecraft has a special pattern to aid the task of relative pose estimation, and that the chaser spacecraft uses a monocular camera as the primary visual sensor. In this sense, the problem falls under the category of cooperative relative navigation in orbit. While existing systems for cooperative localization with fiducial markers allow full 6-DOF pose estimation, the majority of them are not suitable for in-space cooperative navigation (especially when involving a small-size chaser spacecraft), due to their computational cost. Moreover, most existing fiducial-based localization methods are designed for ground-based applications with limited range (e.g., ground robotics, augmented reality), and their performance deteriorates under large scale changes, such as those encountered in space applications. Using an adaptive visual algorithm, we propose an accurate and numerically efficient approach for real-time vision-based relative navigation, especially designed for space robotics applications. The proposed method achieves low computational cost, and high accuracy and robustness, via the following innovations: first, an adaptive visual pattern detection scheme based on the estimated relative pose is proposed, which improves both the efficiency of detection and accuracy of pose estimates; second, a parametric blob detector called Box-LoG is used, which is computationally efficient; and third, a fast and robust algorithm is introduced, which jointly solves the data association and pose estimation problems. In ad-

dition to having an accuracy comparable to state-of-art cooperative localization algorithms, our method demonstrates a significant improvement in speed and robustness for scenarios with large range changes. A vision-based closed-loop experiment using the Autonomous Spacecraft Testing of Robotic Operations in Space (ASTROS) testbed demonstrates the performance benefits of the proposed approach.

1 Introduction

Satellite proximity operations are deemed as an enabling technology that can revolutionize future space operations. The ability to autonomously circumnavigate a target satellite or an asteroid and determine its relative motion is a necessary ingredient to make tasks such as servicing, health-monitoring, surveillance and inspection in orbit or for deep space missions routine [Rekleitis et al., 2007, Sun et al., 2014]. Owing to the large distances involved, human intervention is often not a suitable or timely option. Subsequently, satellite robotic operations require a large degree of autonomy, accuracy and robustness [Flückiger and Utz, 2014]. While relative pose (i.e., position and attitude) estimation can be made easier and more accurate with the use of external aids (ground-based signals or target satellite radio-navigation) or active means (e.g., LiDARs), the same task is more challenging when passive sensors (e.g., vision cameras) have to be used, or when the on-board computational resources of the chaser spacecraft are limited. The latter is typically the case with small chaser satellites. In fact, several space applications (formation flight, persistent Space Situational Awareness (SSA)) call for small satellites to be used in lieu of a larger, monolithic satellite, as it is the current practice. Algorithm development for reliable vision-based relative navigation that is suitable for real-time implementation on such small satellites is currently an open problem.

The motivation behind the proposed work arises from the need for an in-space localization system that achieves good numerical efficiency while, at the same time, provides highly accurate and robust solutions to the relative pose estimation problem for small satellites having limited on-board power and computational resources. Previously, several techniques have been proposed to solve the relative pose estimation problem between two spacecraft in orbit. These techniques either emphasize the sensory data used (GPS in conjunction with IMU data, LiDAR sensing data, etc.), or use additional aids, such as ground station aided relative navigation [DiMatteo et al., 2009, Ruel and Luu, 2010, Kasai et al., 1999, Gaylor and Lightsey, 2003, Ruel et al., 2011]. Their performance may suffer when applied to persistent pose tracking in space over long durations (e.g., IMUs experience drift). LiDAR sensors can be used to provide high accuracy, but LiDAR sensors require a lot of power to operate. An alternative to LiDAR sensors is the use of passive visual sensors that take advantage of the natural light from the Sun to illuminate the target. While the use of passive visual sensors also comes with a unique set of challenges (high contrast in space, continuously and rapidly changing illuminating conditions especially in low Earth orbit, etc), recent developments in visual localization suggest that vision-based relative pose estimation may be a feasible alternative for relative navigation in space. Since vision sensors have become more accurate, smaller, and have low power consumption, they are especially suitable for space applications where the on-board resources (power, computational hardware) are limited and where relative pose maneuvering occurs over long time scales.

Our work falls under the class of vision-enabled cooperative satellite proximity operations [Fehse, 2003]. In the cooperative satellite proximity operations scenario, the objective is to achieve relative navigation with respect to the target satellite, whose motion is not known but it can be inferred by observing a known target pattern attached on the target satellite main body. Although it shares a similar objective with cooperative navigation in other robotic applications, the in-space scenario has particular challenges in the following aspects: (a) it must be efficient both in terms of computation and memory, due to the limited on-board resources; (b) it must be robust with respect to large scale changes in the environment and the unknown status of the target (e.g., target can be in or out of the camera field of view); (c) it requires high accuracy for relative localization; and (d) a high update frequency is required for better closed-loop performance with a pose-tracking controller.

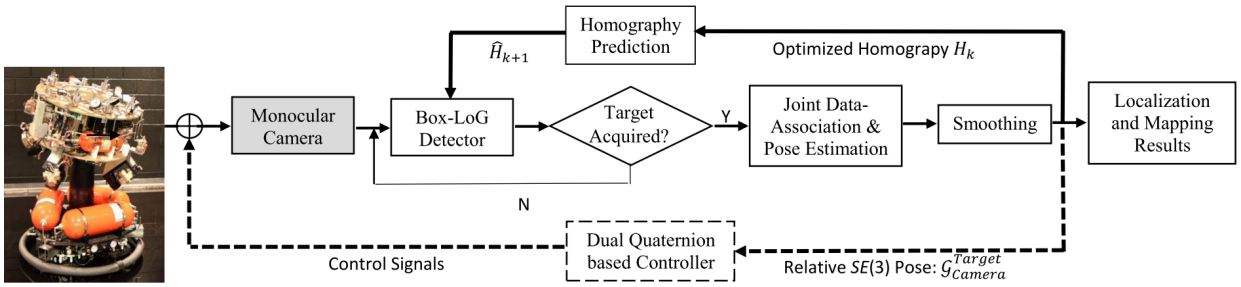


Figure 1: Overall schematic of proposed cooperative navigation system. The system consists of two main feedback loops: one inside the camera localization sub-system, while the other implements the feedback control loop using the inertia-free pose-tracking controller based on dual quaternions from [Filipe and Tsiotras, 2014]. This paper focuses on the localization sub-system, which is independent from the control sub-system. The forward loop of the localization sub-system processes each captured image in real-time. The steps of the localization subsystem are as follows: detection of the target with integral image and Box-LoG kernel; target acquisition; once target is acquired, the relative pose is estimated by jointly solving the data-association and pose estimation problem, and the solution is subsequently optimized via smoothing. In the feedback loop, the predicted homography is fed back to adapt the detector parameters to facilitate detection in the next image frame.

To address the above challenges, this paper proposes a novel closed-loop cooperative navigation approach especially designed for space applications involving small satellites with limited resources. The overall structure of the proposed approach is depicted in Figure 1. The main forward loop implements the camera localization system, which is the main focus of this paper. Note that camera localization is relative with respect to a target whose motion in inertial space may be unknown. The outer loop is utilized to feed back the measured relative pose to an inertia-free pose-tracking controller based on dual quaternions [Filipe and Tsiotras, 2014]. The overall system is experimentally validated using the 5DOF Autonomous Spacecraft Testing of Robotic Operations in Space (ASTROS) facility at the School of Aerospace Engineering of Georgia Institute of Technology. The ASTROS is a realistic experimental platform for testing spacecraft attitude control and similar space proximity operations in a 1-g environment. More details about the capabilities of the ASTROS can be found in [Cho et al., 2009] and [Tsiotras, 2014].

A key contribution of this work is the feedback loop formulation inside the localization system (see Figure 1). The optimized homography is fed to a homography predictor based on a constant motion model. The predicted homography is then decomposed to extract the rotation and scaling effects of the perspective transformation for the next image frame. This is used as prior information to adapt the Box-LoG detector to be used in the next localization iteration. As shown in Figure 1, the camera localization system is structured as a closed-loop system whose intermediate output, the homography from the last frame, is fed back in order to adapt the parameters of the detector. After initialization, and for each captured image from the on-board camera, the proposed Box-LoG detector (see Section 3) is adapted to compensate the perspective transformation between the camera and the target. A subsequent pattern detection step generates the integral image from the raw image, and it convolves the result with a Box-LoG kernel via a set of Dirac delta functions. The use of an integral image along with the Dirac delta functions provides much lower computational complexity compared to the traditional convolution with the original image. The Box-LoG detector determines whether a target is present, and if so, the next step jointly solves the data-association and relative pose estimation problems via robust point-set registration with Gaussian Mixture Model generators. The registration is efficiently optimized over the set of homography maps, and the camera pose is extracted from the optimized homography. The final estimated pose is the output of a smoothing step.

The proposed algorithm addresses some of the challenges of a cooperative spacecraft rendezvous discussed earlier, by incorporating the following advantages: (a) computation and memory efficiency: both the com-

putational and memory complexity are of linear order in terms of the image size; (b) robustness: the ability to deal with the (partial) out-of-view status of the pattern and the adaptivity of distance changes via the multi-scale selection of the pattern, as demonstrated in the experiments; (c) high localization accuracy: the algorithm achieves the same level of accuracy as other state-of-the-art methods, i.e. AprilTag [Olson, 2011]; (d) high update frequency: high update frequency is demonstrated with modest hardware requirements and good overall performance when the algorithm is used in closed-loop with an inertia-free pose-tracking controller based on dual quaternions from [Filipe and Tsiotras, 2014].

This work builds upon our previous work on cooperative navigation, reported in [Zhang et al., 2014], by adding the following specific contributions: First, the method in [Zhang et al., 2014] is extended to achieve real-time performance with limited-capability computational hardware. Second, the original Box-LoG kernel is improved with homography prediction and perspective compensation, both of which improve detection and estimation performance under severe perspective transformations. Third, the controlled closed-loop system is validated through a relative attitude regulation experiment with respect to a moving target using a realistic experimental test platform. Finally, an extensive comparison both in terms of theoretical analysis and using experimental results is performed against a state-of-art cooperative (fiducial) localization method. It should be noted that although the focus of this work is space robotic applications, the same algorithms can be helpful in cooperative, relative navigation for other robotics applications as well, e.g., AUVs, UAVs.

The rest of the paper is structured as follows. Prior related work is discussed in the next section. Section 3 describes in detail the proposed detector, along with the designed target pattern used for cooperative relative navigation. Section 4 outlines the proposed joint pose estimation and data association solution to this problem. Section 5 briefly discusses the smoothing of the estimated relative states. Section 6 covers four sets of experiments: Section 6.1 and Section 6.3 validate the performance of the algorithm using synthetic and field experiments, respectively; Section 6.4 presents the results from closed-loop experiments performed in conjunction with an adaptive pose tracking controller, under the scenario of relative attitude regulation of the chaser spacecraft. A comparison against an existing state-of-art method is presented in Section 6.2. We finally conclude the paper in Section 7 with a summary of contributions and some suggestions for future work.

2 Related Work

The problem of localization and mapping using a camera has been investigated extensively in various fields, including space robotics, AUV (Autonomous underwater vehicle), UAV (Unmanned aerial vehicle), etc. In this section we first provide the reader with a brief literature review of the subject of localization and mapping, emphasizing existing methods most closely related to our problem and the proposed solution approach. Note that although this work focuses on a vision-only system, systems utilizing other sensing methodologies have also been developed for autonomous space rendezvous. For example, the Automated Transfer Vehicle (ATV) achieves localization with a videometer emitting pulsed laser beams, which are further reflected by retroflectors on the target to form unique light patterns [Fehse, 2003]. The Engineering Test Satellite #7 (ETS- VII) from the National Space Development Agency of Japan (NASDA) successfully performed autonomous cooperative rendezvous and docking using RGPS (beyond 500 m from the target), a laser radar (between 2 m and 520 m), and a CCD camera (within 2 m) [Oda, 2001]. A similar sensor, also using lasers, is the Advanced Video Guidance Sensor (AVGS) developed by NASA, which was used in the Demonstration of Autonomous Rendezvous Technologies (DART) and DARPA's Orbital Express programs. AVGS is an enhanced version of the earlier Video Guidance Sensor (VGS), also developed by NASA in 1997, and flown and tested during STS-87 and STS-95 missions [Hintze et al., 2007, Howard and Bryan, 2007].

The first question when designing a vision-only localization system for space robotics applications is the selection between a monocular and a stereo vision system. Stereo systems directly provide depth information, which makes pose estimation much easier. Despite the cost of an additional camera, when the target is close, the large epipolar disparity provides high localization accuracy at a low computational cost; however,

when the target is farther away stereo vision does not seem to offer any significant advantage owing to the small disparity. As a result, stereo-based localization and mapping techniques have been proposed and extensively used in close proximity space robotics applications [Xu et al., 2010, Xu et al., 2009b] and underwater ROVs [Jasiobedzki et al., 2008]. [Howard and Bryan, 2007] reported that AVGS also utilized a stereo-vision system. The two images from the stereo-vision system were used for identifying a known pattern, the retro-reflectors, via image subtraction. The Prisma mission conducted by the Centre National d'Etudes Spatiales (CNES) used a similar approach but with a LED pattern on the target, for autonomous rendezvous with a 50 m to 10 km range [Delpech et al., 2012]. The Synchronized Position Hold Engage Reorient Experimental Satellites (SPHERES) from MIT also utilizes a stereo-vision system for cooperative navigation using visual odometry techniques [Tweddle, 2013]. Although the visual pattern is not pre-stored in the memory of the chaser satellite, a set of textured stickers attached to the target is needed to provide enough visual texture.

Although preferable for most robotics applications, stereo-systems are also more expensive, consume more power, and require precise calibration, compared to monocular systems. Alternative, cheaper approaches tend to use monocular cameras, trading hardware complexity for software complexity. When a stereo system is replaced with a monocular vision system, depth information is lost. Subsequently, relative pose needs to be estimated by tracking landmarks in the environment over consecutive frames. Posterior optimization is also often needed to improve the initial pose estimates. Three problems must be resolved in order to achieve accurate pose estimation using a monocular camera system: (a) feature/landmark detection; (b) data association and pose estimation; and (c) pose filtering.

During the detection phase, the salient features of the target are widely used as detection landmarks, especially in uncooperative scenarios. Typical features include geometric structures such corners [Shi and Tomasi, 1994], blobs [Lindeberg, 1998], or more sophisticated features like SIFT [Lowe, 2004], SURF [Bay et al., 2008], and more recently MROGH [Fan et al., 2012], among others. However, in some space applications the environment may not have sufficient salient features. Moreover, uncooperative methods have scale (depth) ambiguity due to the camera projection transformation. Thus, cooperative vision-based relative navigation methods have been proposed, which assume that some form of a priori knowledge about the target is known. This information is usually the existence of a known pattern on the observed object. Such patterns include special shapes [Saripalli et al., 2003], or especially designed patterns such as self-similar landmarks [Negre et al., 2008], Haar rectangular features [Maire et al., 2009], 2D bar code style patterns [Olson, 2011], rings structures [Velasquez et al., 2009], etc. Detection of these patterns may be computationally costly [Negre et al., 2008], or not robust to large scale changes [Saripalli et al., 2003, Cho et al., 2013, Olson, 2011, Velasquez et al., 2009], or may not provide accurate 6-DOF pose estimation [Maire et al., 2009].

Regarding the data association and pose estimation steps, pose estimation with given corresponding features is widely considered as a solved problem [Hartley and Zisserman, 2000], while data association remains a key problem. Conventionally, data association is solved by matching the feature descriptors under some mapping criterion [Neira and Tardos, 2001], and using a robust statistical framework such as RANSAC [Fischler and Bolles, 1981]. However, these techniques rely on the discriminatory character of the features. Moreover, methods utilizing distinct features require expensive feature matching steps, usually based on image patch matching or feature descriptor matching. For data association without distinct features, some techniques have been proposed based on robust point-set matching [Cho et al., 2013, Wong and Geffard, 2010] or image registration [Karasev et al., 2011]. These techniques are especially useful for cooperative cases in which the features from the target pattern are all similar, such as fiducial dots. Moreover, such approaches avoid expensive matching of descriptors or raw image patch matching, thus reducing the computational overhead.

Typically, each component in a typical monocular vision-based relative pose estimation pipeline operates in an open-loop fashion, with the output of one stage in the pipeline feeding on to the next stage input. There is no feedback of information from a downstream stage to an earlier stage. One of the innovations of the work in this paper is that the proposed processing pipeline includes an information feedback loop, whereby the pose estimates are fed back to the detection step in the pipeline in order to improve target pattern detection reliability, which then impacts future pose estimates.

Among the various existing visual localization system designs, the most relevant to our work are those using fiducial-based 6-DOF localization. Fiducial systems use artificial patterns to keep track of the relative camera/target movement as well as to distinguish between different targets. Although fiducial-based systems usually involve a payload decoding as the last step, all the other steps aim to estimate the 6-DOF camera relative pose history, and thus share the same goal as our work. ARToolkit [Kato and Billinghurst, 1999] and ARTag [Fiala, 2005] are two popular choices for fiducial-based localization, which are widely used in augmented reality applications. ARToolkit detects the target tag by binary thresholding of the image, thus rendering detection sensitive to illumination changes or occlusion. While the ARTag and ARToolkitPlus [Wagner et al., 2008] improve detection robustness with image gradients, these methods are mainly designed for augmented reality applications within a bounded environment, and hence detection is not reliable over longer distances. AprilTag [Olson, 2011] has become a prevalent method for 6-DOF fiducial-based Simultaneous Localization and Mapping (SLAM). This algorithm is designed to be robust and reliable over long distances, while maintaining high accuracy in terms of pose estimation. AprilTag detects the target tag by first computing the image gradient, then clustering the gradient and fitting line segments, and lastly extracting the four-sided regions using a depth-first search. With the quad, the camera poses are further estimated via computing the homography matrix of the encoded points. The author of [Olson, 2011] reports that this approach outperforms previous fiducial-based systems in many aspects. Based on these nice properties, AprilTag is used in this paper to compare against numerical efficiency and accuracy with our approach.

3 Box-LoG Detector and Target Pattern Choices

Relying on a monocular visual sensor for feedback requires algorithms that are invariant or adaptive to imaging variation caused by the unknown, time-varying relative pose between the chase and target satellites. For the scenario considered here, the pattern detection algorithm needs to be invariant to relative orientation about the optical axis, insensitive to the distance from the target, and somewhat robust to the perspective distortion caused by angled views of the pattern. The pattern itself should provide sufficient information to estimate relative pose over several distance scales, including sufficiently close proximity operations, during which only partial views of the pattern may be available. Together, the pattern and detection algorithms should lead to a computationally efficient solution given the hardware limitations of the on-board space electronics. The simplest pattern element fitting these requirements and resulting in an equally simple detection algorithm is a blob (a filled circle). This section details a computationally efficient blob detector and the associated pattern, consisting of nested blob pattern elements, that are designed specifically to work at multiple scales (and hence multiple orders of distance).

3.1 Efficient Detection with the Box-LoG Kernel

Blobs are simple features with mathematically appealing structure across spatial scales [Lindeberg, 1998]. Given an image, blob detection involves analysis of the image Hessian (second-order derivative tensor), with one detection method relying on the determinant of the Hessian and another relying on the trace of the Hessian [Lindeberg, 1998]. Both strategies work well and are optimal for circular blob-like structures, however the simplest of the two is the trace of the Hessian. Feature detection is often combined with a smoothing step and (spatial scale) normalized leading to the Laplacian of Gaussian (LoG) detector, which applies a normalized and smoothed Laplacian operator $\triangle$ to a 2D field. The LoG convolution kernel is defined as

$$\triangle G = \sigma^2 \left(\frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} \right) = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \tag{1}$$

where σ is a function of the blob radius r to detect, $\sigma = r/\sqrt{2}$. For an image I, the operation involves a 2D discrete convolution with the LoG kernel, where the domain of the LoG kernel in (1) is $x, y \in [-R_{\text{LoG}}, R_{\text{LoG}}] \subset \mathbb{Z}$, typically with $R_{\text{LoG}} = \lceil 3\sigma \rceil + 1$ to avoid shift artifacts. Appropriately sized blobs in an image I give large magnitude values in the convolved image $\Delta G * I$.

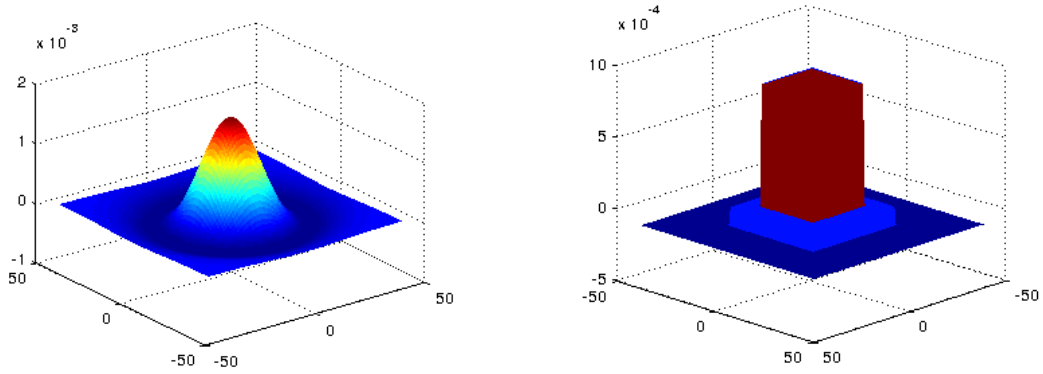


Figure 2: A LoG (left) with $\sigma = 14.1421$ and corresponding Box-LoG (right) kernels. Note that the direction of the z-axis is reversed for better illustration.

Although more complex, the determinant of the Hessian has been popularized by the SURF descriptor [Bay et al., 2008], which employs approximations to the determinant of the Hessian by piecewise constant discrete derivatives in order to achieve efficient blob feature detection. To obtain further computational efficiency when applying the detector at multiple scales, the SURF feature detection algorithm employs integral images based on the identity

$$J = g * I = (g'') * \left(\iint I \right), \tag{2}$$

for any image I convolved with a 2D kernel g [Simard et al., 1999]. Thus the approach when applied to piecewise-constant convolution kernels gives convolution algorithms with linear runtime (in terms of the image size) complexity.

Utilizing a piecewise constant trace of the Hessian approximation one has a lower computational cost, with marginal difference in the output, when compared to the determinant of the Hessian approximation. To ensure that the difference is minimal, the piecewise constant terms must be designed by matching against the equivalent LoG response. To this end, consider an approximation of the LoG kernel $\Delta G(x, y)$ where $x, y \in [-R_{\text{LoG}}, R_{\text{LoG}}]$ with a three box filters such that:

$$\Delta G(x, y) \approx g(x, y) \equiv a_1 H(x, y, R_1) + a_2 H(x, y, R_2) + a_3 H(x, y, R_{\text{LoG}}), \tag{3}$$

where a_1, a_2, a_3 are the coefficients for each box filter to be determined, and $H(x, y, R)$ is the square Heaviside Step Function given by

$$H(x, y, R) = \begin{cases} 1 & \text{if } x \in [-R, R] \wedge y \in [-R, R], \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

To match the response of the LoG kernel for an ideal blob, the approximate version should satisfy the equations

$$\sum_{x,y \in [-R_1, R_1]} \sum \Delta G = \sum_{x,y \in [-R_1, R_1]} \sum g = (a_1 + a_2 + a_3) R_1^2, \tag{5a}$$

$$\sum_{x,y \in [-R_2, R_2]} \sum \Delta G = \sum_{x,y \in [-R_2, R_2]} \sum g = a_1 R_1^2 + (a_2 + a_3) R_2^2, \tag{5b}$$

$$\sum_{x,y \in [-R_{\text{LoG}}, R_{\text{LoG}}]} \sum \Delta G = \sum_{x,y \in [-R_{\text{LoG}}, R_{\text{LoG}}]} \sum g = a_1 R_1^2 + a_2 R_2^2 + a_3 R_{\text{LoG}}^2 = 0. \tag{5c}$$

The last equality is zero because the LoG kernel has zero-mean. The system of equations is linear in the

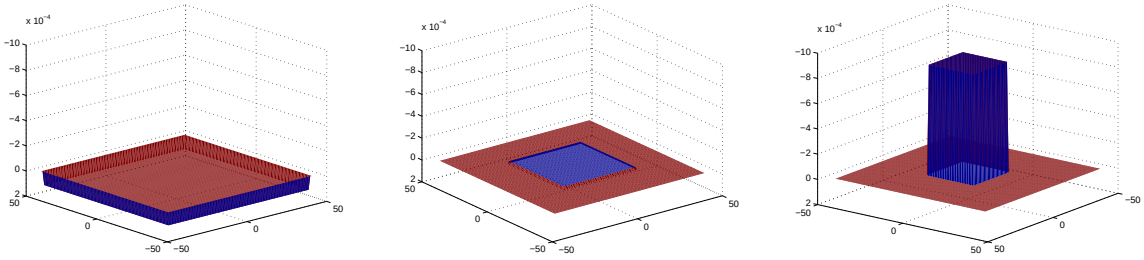


Figure 3: Three rectangular layers of the BoxLoG in Figure 2. Height of outer layer (left) is 1.0967×10^{-4} with $R_{\text{LoG}} = 44$; middle layer (middle) is -2.4868×10^{-5} with $R_2 = 21$; inner layer (right) is -8.7524×10^{-4} with $R_1 = 13$.

coefficients a_1, a_2, a_3, given values of R_1, R_2, and R_{LoG}. Its solution is given by

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} R_1^2 & R_1^2 & R_1^2 \\ R_1^2 & R_2^2 & R_2^2 \\ R_1^2 & R_2^2 & R_{\text{LoG}}^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum\sum_{[-R_1,R_1]} g \\ \sum\sum_{[-R_2,R_2]} g \\ \sum\sum_{[-R_{\text{LoG}},R_{\text{LoG}}]} g \end{pmatrix}. \tag{6}$$

Since R_{LoG} is a function of σ, only the values R_1 and R_2 need to be specified to arrive at the solution. Empirical results show that when R_1 and R_2 satisfy the relations $(R_1 + R_2)/2 = r$ and $R_2 = 2.5R_1$, the approximate LoG gives good detection analogous to the continuous LoG. Solving for R_1 and R_2, yields $R_1 = \lceil \frac{4}{7} r \rceil, R_2 = 2[r] - \lceil \frac{4}{7} r \rceil$, so that the coefficients are completely specified by the detection radius r.

For a given value or r, let the associated *Box-LoG* kernel be defined by the approximate LoG kernel determined by the equations (3) and (6). An example of a LoG kernel and its Box-LoG approximation are depicted in Figure 2. The Box-LoG has several computational advantages over existing approximations. For instance, since computing the trace of a matrix is a simpler operation than computing its determinant and, in addition, the Box-LoG does not require the calculation of mixed second-order derivatives, there will be fewer evaluations of the integral image, compared to [Bay et al., 2008].

The discrete version of the integral image is defined to be:

$$S(x, y) = \sum_{x' \le x} \sum_{y' \le y} I(x', y'). \tag{7}$$

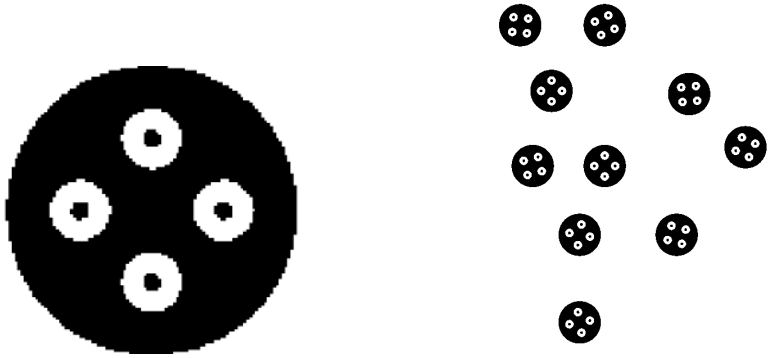
The second derivative of Box-LoG consists of a linear combination of eight Dirac delta functions, leading to eight evaluations of S for the Box-LoG computation $J = g * I$, as follows

$$\begin{aligned} J(x, y) = \sum_{i=0}^{1} \sum_{j=0}^{1} (-1)^{i+j} (a_1 - a_2) S\left(x + (-1)^i R_1\,,\, y + (-1)^j R_1\right) \\ + (-1)^{i+j} (a_2 - a_3) S\left(x + (-1)^i R_2\,,\, y + (-1)^j R_2\right). \end{aligned} \tag{8}$$

After computing the response image J for a discrete quantity of radius scales (octaves in the computer vision parlance), the blob detection process then thresholds the response magnitudes followed by non-maximum suppression (dark blobs give positive extrema and light blobs give negative extrema).

3.2 Landmark Pattern Design

The use of circular pattern elements (e.g., blobs) provides rotational invariance about the optical axis and relative insensitivity to view-point deviations from the optical axis during Box-LoG detection. What remains is to define a pattern consisting of blobs that fulfills the remaining requirements. Converting the remaining specification into a list of desired properties for the target pattern results in the following list:



(a) Chosen pattern element (b) Candidate pattern

Figure 4: Pattern element and landmark pattern for cooperative tracking.

i) pattern elements at multiple scales, for robustness to scale changes;

ii) co-planarity, for rapid pose estimation through homographic geometry;

iii) sufficient number of pattern elements, for well-posed pose estimation; and

iv) an asymmetric and non-collinear topology, to avoid degeneracy and pose ambiguity due to rotations or perspective foreshortening.

A pattern element or marker feature that achieves the first specification consists of nested blobs at different scales and complementary contrasts (dark on light vs. light on dark). As shown in Figure 4, the marker design is such that the circle radius at one blob scale is 4.5 times greater than that of the next nested smaller scale. This factor ensures that the nested blobs can be arranged such that a properly adapted Box-LoG filter centers exactly on a blob without getting response interference from a neighboring blob scale. When combined with the multi-scale blob detector from Section 3.1, the blobs at a given scale can be robustly tracked until the next scale is identified (about two octaves later and with the opposite contrast). Further, the blob markers at the three different scales provide three detection modes determined by the relative distance between the target and the camera. During the experiments described in Section 6, the detection system switches the target blobs from the current scale to the next smaller scale when the image radius of the current scale becomes larger than 10 pixel units, except for the smallest scale.

To fulfill the last three properties, the pattern should have, at a minimum, three non-collinear pattern elements on a planar surface (for homography-based pose estimation). To be robust to partial occlusions or to pattern elements leaving the image frame, at least five markers are used [Nistér, 2004] and arranged asymmetrically. Moreover, each marker should be at least one diameter in distance away from other markers to avoid false positive detections (in the area between two markers). A pattern with these characteristics is shown in Figure 4(b). It consists of ten pattern elements randomly scattered on a square area, each rotated at a random angle[1]. During the initial pattern detection phase, when more than 4/5 of the pattern is detected, then the pattern is considered to be acquired. During tracking, the Box-LoG detection radius is specified according to the relative pose between the chase and target satellites. When the relative distance is large, the larger markers are set to be detected. As the camera gets closer to the target, the smaller nested markers are set to be detected. The following section describes the active adaptation process used in order to optimize the visual processing pipeline and improve the relative pose feedback to the pose tracking controller.

Note that in this work, severe illumination changes are not considered. Mild illumination changes are mitigated by the complementary contrast of the nested blobs.

[1]The pattern and quantity of pattern elements is a design choice.

3.3 Adaptive Compensation to Perspective Imaging Distortion

The image formed by a circular marker as seen through a (perspective) camera depends on the intrinsic camera parameters, the marker's actual radius on the tag, and the camera to marker distance. When the marker's actual radius is fixed, there is an inverse linear relationship between the image radius and the camera-to-marker distance. When the vector normal to the planar pattern is not aligned with the camera's optical axis, the blob-like image is warped by a perspective transformation. Under a severe perspective transformation, the square-shape Box-LoG kernel may fail to detect the pattern. Relative pose information available during the closed-loop engagement scenario can be used to actively modify the parameters of the Box-Log detection strategy, as well as to pre-process the image for optimal detection.

The main parameter of the Box-LoG algorithm is the expected detection radius. During tracking, the radius is known from the initial pattern detection phase (which cycles through the various radii until the target is acquired). Thus, as part of the processing pipeline, the Box-LoG kernel radius is adapted via feedback of the estimated target position from the previous frame (lower box of Figure 1), based on the inverse distance relationship:

$$r_{k+1} = \frac{\lambda}{\sqrt{\tilde{x}_k^2 + \tilde{y}_k^2 + \tilde{z}_k^2}}, \tag{9}$$

where λ is a constant determined by the target marker's world radius (a known constant) and the intrinsic camera parameters, $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ is the relative position of the camera with respect to the target center in the k-th frame, and r_{k+1} is the detection radius estimate for the $(k+1)$-th frame.

Compensation for perspective warping effects is typically achieved by de-warping the image according to the inverse of the perspective transformation. Computationally this involves generating the warp function for each pixel and then using interpolation on the image to apply the warp. However, such a method is expensive both in terms of the number of required computations and memory. To efficiently tackle this problem, we propose to approximate the perspective transformation by a computationally cheap integer-based image rotation operation, followed by a modification of the Box-LoG so as to always be of rectangular shape. Together, these two steps compensate for the perspective warp while minimizing the computational cost.

The rotated image is efficiently generated by remapping the set of horizontal lines of the original image to the set of pixelized, integer-based parallel lines of the rotated image, so that the two sets differ by the rotation angle θ. The pixel positions along each remapped line are determined by Bresenham's line drawing algorithm [Bresenham, 1965]. Pixels in the remapped image that do not originate in the sensed image are set to default background values. The computational complexity of the rotation operation is linear with respect to the image area, both in terms of the number of computations involved and memory.

The Box-LoG kernel applied to the rotated image is further modified by changing the height-to-width ratio κ of the kernel. When $\kappa \neq 1$, each level of the Box-LoG is of rectangular shape instead of a square shape, specifically,

$$g(x,y) \equiv a_1 H_{\text{rect}}(x, y, R_1^x, R_1^y) + a_2 H_{\text{rect}}(x, y, R_2^x, R_2^y) + a_3 H_{\text{rect}}(x, y, R_{\text{LoG}}^x, R_{\text{LoG}}^y), \tag{10}$$

where a_1, a_2, a_3 are solved from equation (6) and $H_{\text{rect}}(x, y, R^x, R^y)$ is the rectangular Heaviside Step Function given by

$$H_{\text{rect}}(x, y, R^x, R^y) = \begin{cases} 1 & \text{if } x \in [-R^x, R^x] \wedge y \in [-R^y, R^y], \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

The shape of Box-LoG is then adapted according to κ such that $R_1^x/R_1^y = R_2^x/R_2^y = R_{\text{LoG}}^x/R_{\text{LoG}}^y = \kappa$. When $\kappa > 1$, $R_{\text{LoG}}^x = \lceil 3r_{k+1}/\sqrt{2} \rceil + 1, R_1^x = \lceil \frac{4}{7} r_{k+1} \rceil, R_2^x = 2[r_{k+1}] - \lceil \frac{4}{7} r_{k+1} \rceil$, and when $\kappa < 1$, $R_{\text{LoG}}^y = \lceil 3r_{k+1}/\sqrt{2} \rceil + 1, R_1^y = \lceil \frac{4}{7} r_{k+1} \rceil, R_2^y = 2[r_{k+1}] - \lceil \frac{4}{7} r_{k+1} \rceil$.

The rotation angle θ and Box-LoG rectangular ratio κ are obtained from the predicted homography $\hat{H}_{k+1}$

for the $(k+1)$-th frame under a constant transformation model:

$$\hat{H}_{k+1} = \hat{H}_{k\to(k+1)} H_k = H_{(k-1)\to k} H_k = H_k H_{k-1}^{-1} H_k, \tag{12}$$

where $H_{(k-1)\to k}$ is the homography mapping points from $(k-1)$-th image to k-th image. The second equality in (12) is due to the constant velocity model. Let the estimate in (12) be written as follows

$$\hat{H}_{k+1} = \begin{pmatrix} A & b \\ \mathbf{0}_{1\times 2} & 1 \end{pmatrix} \tag{13}$$

for some matrix $A \in \mathbb{R}^{2\times 2}$ and $b \in \mathbb{R}^2$. The rotation angle and rectangular ratio are extracted via the singular value decomposition (SVD) of A

$$A = U_A \Sigma_A V_A^{\mathsf{T}}, \tag{14}$$

where $U_A, V_A \in \mathbb{R}^{2\times 2}$ are unitary matrices and $\Sigma_A \in \mathbb{R}^{2\times 2}$ is a diagonal matrix. Note that the SVD of a 2×2 matrix can be computed in closed form. The matrix U_A in (14) is the rotation transformation and Σ_A is the scaling transformation. Thus

$$\theta = \operatorname{acos}(U_{A\,1,1}) \qquad \text{and} \qquad \kappa = \frac{\Sigma_{A\,1,1}}{\Sigma_{A\,2,2}}. \tag{15}$$

In practice, θ need not be computed since the image boundaries are transformed using the (rotation) matrix U_A. Those transformed coordinates then define the parallel lines to follow. With the adapted Box-LoG kernel, the integral image of the de-rotated image is convolved with the Dirac delta functions, and any blobs with the targeted radii in the image are detected. A non-maximum suppression is then performed to refine these detected areas, and sub-pixel detection results are generated by computing the center of mass in each of the non-maximum suppressed areas. The original image coordinates are obtained by rotating the final detected positions.

Figure 5 illustrates the results of performing a Box-LoG detection with perspective compensation. The input image in Figure 5(a) is under a perspective transformation whose effective homography is

$$H = \begin{pmatrix} 0.7507 & 0.3752 & 0 \\ 0.0801 & 0.5708 & 93.8600 \\ 0 & 0 & 1 \end{pmatrix}. \tag{16}$$

By decomposing the matrix A of the homography matrix via SVD, the rotation and scaling matrices are computed as follows

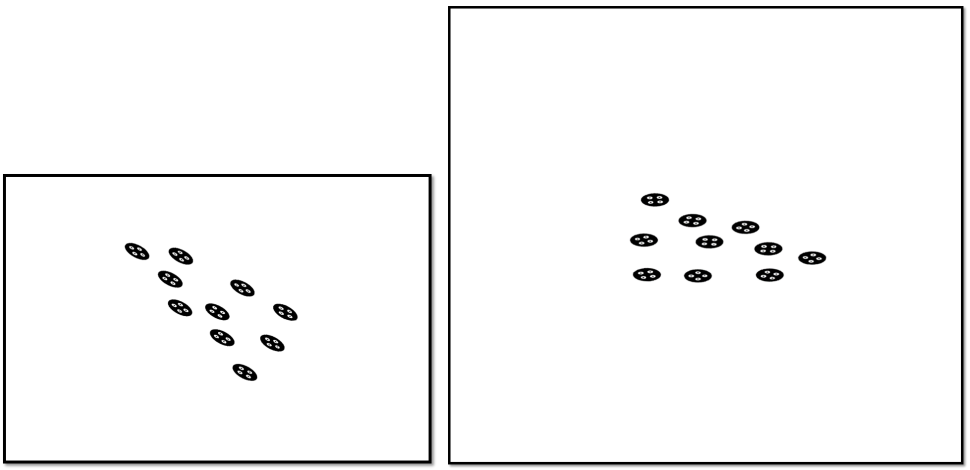
$$U_A = \begin{pmatrix} -0.8835 & -0.4684 \\ -0.4684 & 0.8835 \end{pmatrix}, \qquad \Sigma_A = \begin{pmatrix} 0.9218 & 0 \\ 0 & 0.4322 \end{pmatrix}, \tag{17}$$

which means the image needs to be rotated clockwise by 152.07 deg. The rotated image via Bresenham's line iteration is depicted in Figure 5(b), where the image has been expanded to fit the rotated image area and the unmapped pixels have been filled in with white. The original image is 853×569 pixels while the rotated image is 1020×902 pixels. Convolution is performed using the κ-adapted Box-LoG to generate the response seen in Figure 5(c). The detected blobs' sub-pixel positions are extracted and rotated back to the original orientation as per Figure 5(d).

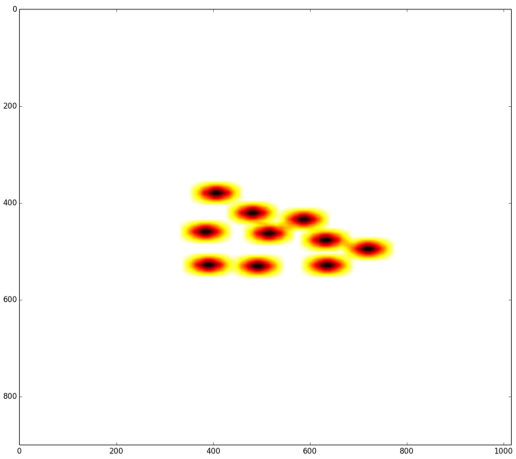
The detection algorithm is summarized in Algorithm 1, along with the corresponding computational complexity for each step. The dominant steps are the image rotation, image integral and fast convolution with the Dirac delta functions. Because all of these steps are of linear order with respect to the image size, the total complexity of the detection algorithm is also of linear order with respect to the image size.

4 Joint Pattern Tracking and Pose Estimation

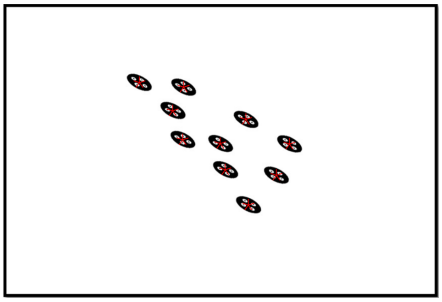
Pose estimation occurs between consecutive frames using the pixel locations of the detected markers. To be robust to false-positive and true-negatives, rather than imposing or seeking one-to-one point correspondences



(a) Image under perspective transformation. (b) De-rotated image using integer image rotation.



(c) Box-LoG response with scaling adaptation.



(d) Final extracted pattern blob positions (red stars).

Figure 5: Perspective compensation process within Box-LoG detector for a target pattern under perspective transformation.

between two consecutive images, this section describes a homography-seeking robust point set registration algorithm. The algorithm attempts to align the two point sets without imposing explicit correspondences. The final alignment provides the correspondences, and hence the required pattern tracking.

4.1 Homography Map

Denote the markers' (homogeneous) locations in the previous image frame as $\boldsymbol{v}_i \in \mathbb{R}^2$, the markers' (homogeneous) locations on the current image frame as $\boldsymbol{u}_i \in \mathbb{R}^2$, for $i = 1 \dots n_m$, and let the (homogeneous) 3D positions $\boldsymbol{X}_i \in \mathbb{R}^3$ of the markers on the pattern plane be given, such that $\boldsymbol{\pi}^\mathsf{T} \boldsymbol{X_i} = 0$, where $\boldsymbol{\pi} = (\boldsymbol{\zeta}^\mathsf{T}, 1)^\mathsf{T}$, $\boldsymbol{\zeta} \in \mathbb{R}^3$, for $i = 1 \dots n_m$, where n_m is the number of markers and $\boldsymbol{\zeta}$ is the normal to the pattern plane. For simplicity, let the previous camera pose be the identity pose, i.e., having camera projection matrix $P_{\boldsymbol{v}} = [I \,|\, 0]$.

Assume that the camera moves rigidly from the previous to the current frame, and hence its motion is given

Algorithm 1: Box-LoG detection with perspective compensation. Final complexity: $\mathcal{O}(NM)$

Data: Input binary image frame $I_{k+1} \in \mathbb{R}^{N \times M}$, previous estimated homography H_k, H_{k-1}, previous estimated pose g_k

Result: Detected blobs sub-pixel positions $\{y_{k+1}^{(i)}\}$

while $I_{k+1} \neq \emptyset$ **do**

- $\hat{H}_{k+1} \leftarrow$ Homography_Prediction(H_k, H_{k-1}) ; // Equations 12
- $\theta, \kappa \leftarrow$ Homography_Decompose$(\hat{H}_{k+1})$; // Equations 14, 15
- $g(x, y) \leftarrow$ BoxLoG_Kernel_Generation(κ, g_k) ; // Equations 3 to 6
- $I'_{k+1} \leftarrow$ Fast_Image_Rotation(I_{k+1}, θ) ; // $\mathcal{O}(NM)$, based on Bresenham's line iteration
- $S'_{k+1} \leftarrow$ Image_Integration(I'_{k+1}) ; // $\mathcal{O}(NM)$
- $J'_{k+1} \leftarrow$ Dirac_Delta_Function_Convolution(S'_{k+1}, g) ; // $\mathcal{O}(NM)$
- $\{y'^{(i)}_{k+1}\} \leftarrow$ Detection_Refinement(J'_{k+1}) ; // Non-maximum suppresion and sub-pixel accuracy detection, $< \mathcal{O}(NM)$
- $\{y^{(i)}_{k+1}\} \leftarrow$ Pixel_Position_Rotation $(\{y'^{(i)}_{k+1}\}, -\theta)$

by the rigid transformation

$$g_{\boldsymbol{v}}^{\boldsymbol{u}} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}, \tag{18}$$

where R is the rotation and T is the translation of the camera between the two frames. The camera projection on the current frame is then given by

$$P_{\boldsymbol{u}} = [R \,|\, T]. \tag{19}$$

Since $\boldsymbol{v} = [I|0]\boldsymbol{X}$ the back-projecting ray of $\boldsymbol{v}$ is $\boldsymbol{X}_{\boldsymbol{v}} = (\boldsymbol{v}^\mathsf{T}, \rho)^\mathsf{T}$, where ρ is the distance of $\boldsymbol{X}_{\boldsymbol{v}}$ to the camera center and, moreover, $\boldsymbol{X}_{\boldsymbol{v}}$ lies on plane $\boldsymbol{\pi}$, i.e. $\boldsymbol{\pi}^\mathsf{T}\boldsymbol{X}_{\boldsymbol{v}} = 0$. Combining these expressions yields $\boldsymbol{X}_{\boldsymbol{v}} = (\boldsymbol{v}^\mathsf{T}, -\boldsymbol{\zeta}^\mathsf{T}\boldsymbol{v})^\mathsf{T}$. If $\boldsymbol{v}, \boldsymbol{u}$ correspond to the same 3D points in the world frame, i.e., if $\boldsymbol{X}_{\boldsymbol{u}} = \boldsymbol{X}_{\boldsymbol{v}}$, then

$$\boldsymbol{u} = P_{\boldsymbol{u}}\boldsymbol{X}_{\boldsymbol{u}} = [R|T]\boldsymbol{X}_{\boldsymbol{v}} = [R|T](\boldsymbol{v}^\mathsf{T}, -\boldsymbol{\zeta}^\mathsf{T}\boldsymbol{v})^\mathsf{T} = (R - T\boldsymbol{\zeta}^\mathsf{T})\boldsymbol{v} \triangleq H\boldsymbol{v}. \tag{20}$$

Since the two views are of points in the same plane, the previous equation shows that the homography map relates corresponding points between consecutive frames (i.e., it maps $\boldsymbol{v}_i$ to $\boldsymbol{u}_i$) [Hartley and Zisserman, 2000]. Note from (20) that the homography map is completely characterized by the transformation from the previous pose to the current pose and by the plane's normal vector.

When the point correspondences and the camera intrinsic matrix are known, the homography, and ultimately the rigid motion transformation matrix $g_{\boldsymbol{v}}^{\boldsymbol{u}}$, is computable. The computation of the transformation matrix from the homography matrix utilizes the constraint that R is a unitary matrix and thus $R^3 = R^1 \otimes R^2$, where R^i is the i-th column of the rotation matrix [Xu et al., 2009a]. Conversely, when the homography is known, then the points can be placed into correspondence and tracked. The problem arises when neither of them is known, and the point sets have extra or missing elements (due to false positive or true negative detections). To handle these uncertainties, the next section jointly solves the pose estimation and point tracking problems using robust point-set registration.

4.2 Robust Point-Set Registration

In robust point-set registration, each image point set $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^{|\mathbf{U}|}$ and $\mathbf{V} = \{\mathbf{v}_j\}_{j=1}^{|\mathbf{V}|}$ of two consecutive images of potentially different cardinality, generates a Gaussian Mixture Model (GMM), the first of which is also transformed by an unknown homography map H. Point-set registration is performed by minimizing the L_2 distance of the GMMs [Jian and Vemuri, 2005, Jian and Vemuri, 2011]. Normally, the minimization is performed over the space of rigid or affine transformations (plus possibly a parameterized model of non-affine deformations). However, in this work the minimization is performed over the space of homographic maps.

Recall that the GMM generator for a set of points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{|\mathbf{X}|}$ is

$$\Phi\left(\mathbf{x}\,;\,\mathbf{X}\right) = \frac{1}{|\mathbf{X}|}\sum_{i=1}^{|\mathbf{X}|}\mathcal{N}\left(\mathbf{x}\,;\,\mathbf{x}_i,\Sigma\right), \tag{21}$$

where $|\mathbf{X}|$ is the cardinality of the set $\mathbf{X}$, and $\mathcal{N}\left(\cdot\,;\,\mathbf{x}_i,\Sigma\right)$ is the multi-variate normal distribution with mean $\mathbf{x}_i$ and (constant) covariance Σ (here a diagonal matrix with equal variances). When the homography map is included in the GMM generator as a parameter, then

$$\Phi\left(\mathbf{x}\,;\,\mathbf{X},H\right) = \frac{1}{|\mathbf{X}|}\sum_{i=1}^{|\mathbf{X}|}\mathcal{N}\left(\mathbf{x}\,;\,A\mathbf{x}_i+b,A\Sigma A^{\mathsf{T}}\right), \tag{22}$$

given that the homography map of an image point $\mathbf{x}\in\mathbb{R}^2$ is $H(\mathbf{x}) = A\mathbf{x}+b$. Given two points sets $\mathbf{U}$ and $\mathbf{V}$, and a homography map H, the registration error is defined by the L_2 distance of the generated GMMs as

$$\mathrm{dist}(\Phi\left(\cdot\,;\,\mathbf{U},H\right),\Phi\left(\cdot\,;\,\mathbf{V}\right)) \triangleq \int\left(\Phi\left(\mathbf{x}\,;\,\mathbf{U},H\right)-\Phi\left(\mathbf{x}\,;\,\mathbf{V}\right)\right)^2\mathrm{d}\mathbf{x}. \tag{23}$$

The multi-variate Gaussian distribution obeys the identity

$$\int\mathcal{N}\left(\mathbf{x}\,;\,\boldsymbol{\mu}_1,\Sigma_1\right)\mathcal{N}\left(\mathbf{x}\,;\,\boldsymbol{\mu}_2,\Sigma_2\right)\mathrm{d}\mathbf{x} = \mathcal{N}\left(0\,;\,\boldsymbol{\mu}_1-\boldsymbol{\mu}_2,\Sigma_1+\Sigma_2\right). \tag{24}$$

As a result, $\mathrm{dist}(\Phi\left(\cdot\,;\,\mathbf{U},H\right),\Phi\left(\cdot\,;\,\mathbf{V}\right))$ can be computed in closed-form as follows

$$\begin{aligned}\mathrm{dist}(\Phi\left(\cdot\,;\,\mathbf{U},H\right),\Phi\left(\cdot\,;\,\mathbf{V}\right)) =& \frac{1}{|\mathbf{U}|^2}\sum_{i=1}^{|\mathbf{U}|}\sum_{j=1}^{|\mathbf{U}|}\mathcal{N}\left(0\,;\,A(\boldsymbol{u}_i-\boldsymbol{u}_j),2A\Sigma A^{\mathsf{T}}\right)\\ &-2\frac{1}{|\mathbf{U}||\mathbf{V}|}\sum_{i=1}^{|\mathbf{U}|}\sum_{j=1}^{|\mathbf{V}|}\mathcal{N}\left(0\,;\,H(\boldsymbol{u}_i)-\boldsymbol{v}_j,A\Sigma A^{\mathsf{T}}+\Sigma\right)\\ &+\frac{1}{|\mathbf{V}|^2}\sum_{i=1}^{|\mathbf{V}|}\sum_{j=1}^{|\mathbf{V}|}\mathcal{N}\left(0\,;\,\boldsymbol{v}_i-\boldsymbol{v}_j,2\Sigma\right).\end{aligned} \tag{25}$$

The homography is obtained by minimizing $\mathrm{dist}(\Phi\left(\cdot\,;\,\mathbf{U},H\right),\Phi\left(\cdot\,;\,\mathbf{V}\right))$ over H,

$$H = \arg\min_{H}\mathrm{dist}(\Phi\left(\cdot\,;\,\mathbf{U},H\right),\Phi\left(\cdot\,;\,\mathbf{V}\right)) \tag{26}$$

After finding H, two points $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are considered to be in correspondence if they have minimal distance compared to all other possible correspondences, and the minimizing distance is below a given threshold. Minimization of (25) is performed iteratively through gradient descent. Note that the last term in (25) is constant, having no effect on the optimization, and thus it can be removed from the computations.

5 Smoothing the Pose Estimates

While the pose estimates are optimized for the current observations conditioned on the previous observations (Section 4.2), they are not optimized temporally over all observations (e.g., they are not filtered). The reason why smoothing may be preferable to filtering is beyond of scope of this paper. The interested readers can refer to [Strasdat et al., 2012]. For vision-based measurements, temporal smoothing is performed by minimizing the image re-projection errors, given the set of pose estimates and homographic mappings to date.

Denote by $\mathcal{G}_t \triangleq \{g_\tau\}_{\tau\le t}$ the set of camera poses up to time instant t and by $\mathcal{Z}_t \triangleq \{\boldsymbol{\xi}_\tau\}_{\tau\le t}$ the collection of measurements up to time t, where $\boldsymbol{\xi}_t$ consists of the points $\{\boldsymbol{u}_{i,t}\}_{i=1}^{n_m}$, where $\boldsymbol{u}_{i,t}$ denotes the measurement

of $\boldsymbol{u}_i$ at time t. Let $\mathcal{L}_t \triangleq \{\boldsymbol{l}_{\alpha_\tau(\cdot)}\}_{\tau \leq t}$ be the set of target pattern landmarks up to time instant t, where $\alpha_t(\cdot)$ is a time-dependent association function that matches a measurement index to a landmark index at time t (this function is instantiated when the pattern is detected and maintained during marker tracking). Define the measurement function $h(g, \boldsymbol{l})$ to be the perspective camera projection, mapping a 3D point $\boldsymbol{l}$ of the target pattern landmark to a 2D image coordinate at camera pose g. Given a measurement and landmark association, the image re-projection error for measurement index i at time t is

$$\boldsymbol{\varepsilon}_{i,t} = h(g_t, \boldsymbol{l}_{\alpha_t(i)}) - \boldsymbol{u}_{i,t}. \tag{27}$$

Assuming Gaussian measurement noise, the distribution of the measurement given the landmark positions is

$$P\left(\boldsymbol{u}_{i,t} | g_t, \boldsymbol{l}_{\alpha_t(i)}\right) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{\varepsilon}_{i,t}\|^2_\Sigma\right). \tag{28}$$

where Σ is the covariance matrix of pixel noise as in Section 4.2. Let now $\Theta \triangleq (\mathcal{G}_t, \mathcal{L}_t)$ denote the collection of the unknown camera poses and landmarks observed up to time t, and model the system using a factor graph [Kschischang et al., 2001]. In our case, no odometry information is available because the target's motion is unknown with respect to the inertial frame. Therefore, there are no factors encoding the prediction model. Using the factorization property of factor graphs, the joint probability of the random variables Θ is [Dellaert and Kaess, 2006]

$$P(\Theta) \propto \left(\prod_t \varphi_t(\boldsymbol{\theta}_t)\right)\left(\prod_{t,j} \psi_{t,j}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_j)\right), \tag{29}$$

where the t index runs over the variables in $\mathcal{G}_t$, j index runs over the variables in $\mathcal{L}_t$, and the potentials $\varphi_t(\boldsymbol{\theta}_t)$ encode the prior estimate at $\boldsymbol{\theta}_t \in \Theta$, and the pairwise potentials $\psi_{t,j}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_j)$ encode information between two factors (here, a camera pose and a landmark). Using this information, the potentials are

$$\varphi_t(\boldsymbol{\theta}_t) \propto P(g_t) \tag{30}$$

$$\psi_{t,j}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_j) \propto P(\boldsymbol{u}_{\alpha_t^{-1}(j),t} | g_t, \boldsymbol{l}_j). \tag{31}$$

For the second set of potentials, $\psi_{t,j}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_j)$, the potential (and hence factor graph edge) does not exist when the inverse is not defined for a given (t, j) (i.e., the landmark was not seen). The maximum a posteriori (MAP) estimate is

$$\hat{\Theta} = \arg\max_\Theta P(\Theta | \mathcal{Z}_t) = \arg\max_\Theta P(\Theta, \mathcal{Z}_t) = \arg\min_\Theta(-\log P(\Theta, \mathcal{Z}_t)). \tag{32}$$

Since the information is arriving sequentially in time, the incremental smoothing method [Kaess et al., 2008, Kaess et al., 2012] is used for optimizing the pose estimates. We use the GTSAM library [Dellaert, 2012], written by the authors of the above references, to implement the incremental smoothing step.

6 Experiments and Discussion

This section evaluates the processing pipeline described in the previous sections, and depicted in Figure 1, on both synthetic and actual relative motion scenarios. Accuracy is evaluated for both position and orientation separately. Position accuracy is measured in terms of a percentage using the relative norm of the relative position error. Specifically, let $\tilde{X}$ be the estimated camera position, X be the ground-truth camera position, and X_T be the center of the target, all in the world-frame. The position accuracy used is then $100\|\tilde{X} - X\|_2 / \|X - X_T\|_2$. The orientation accuracy is given by the error of the estimated camera orientation computed via the norm of the logarithm on $SO(3)$ converted to degrees. Specifically, suppose that $\tilde{R}$ is the estimated orientation and R is the ground-truth orientation, then the error is

$$\mathcal{E}_{SO(3)} = \frac{180}{\pi}\left\|\left(\log_{SO(3)}(\tilde{R}^\intercal R)\right)^\vee\right\| \tag{33}$$

where the "unhat" operation $(\cdot)^\vee$ maps a 3×3 skew-symmetric matrix to a vector.

Experimental validation is performed on the 5DOF spacecraft simulator testbed (Autonomous Spacecraft Testing of Robotic Operations in Space - ASTROS) at Georgia Tech, which is depicted in Figure 6. The spacecraft (seen in Figure 6(b)) has a lower stage (the pedestal) and an upper stage (main spacecraft bus). The lower stage consists of four high-pressure air storage vessels, three linear air-bearing pads, a hemi-spherical air-bearing cup (connecting the lower and upper stages), along with dedicated electronics and power supply. When placed on the flat epoxy floor, of dimensions approximately 14 ft $\times$ 14 ft, with the air pads activated, the spacecraft experiences almost friction-free conditions. The main structure of the ASTROS is the upper stage, whose operational characteristics can be found in [Cho et al., 2009]. The upper stage represents a typical spacecraft "bus" and is made of a two-level brass structure that is supported on a hemi-spherical air bearing allowing rotation of the upper stage with respect to the supporting pedestal about all three axes (± 30 deg about the x and y axes and a full rotation about the z axis).

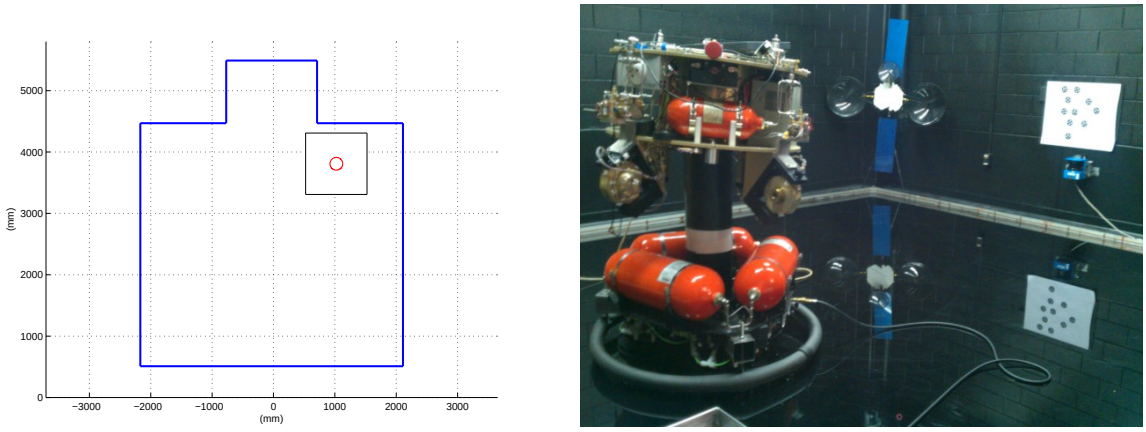
For image capturing, a CCD camera (TMS-730p by Pulnix) mounted on the test bed is connected to a PC-104 Meteor II-Morphis frame grabber (MOR+/2VD/J2K by Matrox Imaging) with a digitizer resolution of 640×480. For on-board image processing, there is a PC-104-Plus computer running Ubuntu 10.04, with a 1.8 GHz Pentium M CPU, 1 GB of RAM, and a 64 GB Compact Flash drive. All vision code is implemented in C++ without SIMD optimization due to the limitations of the on-board CPU. A six camera Vicon^{TM} system captures the ground-truth pose of the upper stage of the platform, which is related to the camera frame by a rigid transformation estimated as part of system calibration, and the target pattern pose.

During closed-loop operation, a second on-board computer runs the controller. This computer is an ADLink NuPRO-775 Series PC with an Intel Pentium III 750 MHz CPU, 128 MB DRAM, and 128 MB disk-on-chip. The two on-board computers communicate via UDP protocol. The controller is implemented as a Simulink model, shown in Figure 7, and then is uploaded to the platform using MATLAB's xPC Target environment. Three Variable-Speed Control-Moment Gyroscopes (VSCMG) are used to control the attitude of the platform [Cho et al., 2009]. A VSCMG can function either as a reaction wheel or as a control moment gyro; attitude of the platform can be controlled by changing the angular speed of the wheel inside the gimbal of the VSCMG or by rotating the gimbal itself. The control torque calculated by the controller is allocated between the three VSCMGs following the approach in [Yoon and Tsiotras, 2002]. A set of 12 thrusters provides translational motion.

6.1 Synthetic Image Experiments of Camera Localization

We first validated our algorithm using synthetic image sequences. The benefits of synthetic experiments are: first, the experiments are fully controlled with accurate ground-truth camera trajectories and camera intrinsic parameters; second, in the synthetic environment we can test scenarios involving camera movements that cannot be tested in field tests due to the degree-of-freedom restrictions (no translation along the vertical axis) of the platform. In the experiments, a 3D virtual reality environment with the designed target is first simulated. Then, a simulated camera moves along a designated trajectory capturing images of the target according to a pinhole camera projection model. The focal length of the simulated camera is $1,388$ mm and the resolution is 1082×722 pixels. The algorithm was tested on synthetic images and the results were evaluated. In this experiment, there is no distortion in the camera projection and no noise in the camera movement.

Four trajectories were simulated. The trajectories and the (measurement) camera poses for each simulation are shown in the first row of Figure 8. Each simulated trajectory consists of motion primitives (straight motion, camera rotation, circular motion, etc.) that a normal engagement scenario might consist of. Some of these motion primitives have different perspective imaging properties that influence the relative position and orientation estimates in different ways. In the second scenario the camera performed a pure rotation from 0 to 360 deg counterclockwise with respect to its optical axis with constant angular velocity. Most motions also involve large perspective changes over the course of the trajectory which also tests the (adaptive) pattern



(a) Dimension of the field for testing

(b) Setup of the platform and the target

(c) Vicon^{TM} setup

Figure 6: Experimental testbed. Figure (a) depicts the actual dimensions of the testing arena, where the blue lines are the boundary of the area in which the ASTROS can reach. Figure (b) shows a picture of the experimental platform. The target is attached to the wall and the ASTROS can move freely on the floor. Figure (c) illustrates the Vicon^{TM} setup, with the marker cameras (green), platform frame (red), target pattern frame (blue) and field floor (orange).

detection algorithm.

In Figure 8, the second and third columns contain the graphs of the relative position accuracy (percentage) and the orientation error (degrees); lower is better in both cases. After the smoothing step both estimates have good accuracy: the smoothed relative position estimates are all within 3% relative error, and relative orientation error is less than 0.2 deg. These results demonstrate that the proposed algorithm can detect the pattern, estimate the relative poses accurately, and adapt the detection scale accordingly.

6.2 Comparison with Existing Fiducial Marker System

The prevailing approaches to tag-based (e.g., planar known patch marker) localization are edge- or line-based and often encode marker identity information as part of the tag (called the data payload) [Olson, 2011, Wagner et al., 2008, Fiala, 2005]. They have been used for relative pose estimation [Olson et al., 2012] and global localization [Lorenz et al., 2012]. This section includes a comparison of the proposed target pattern

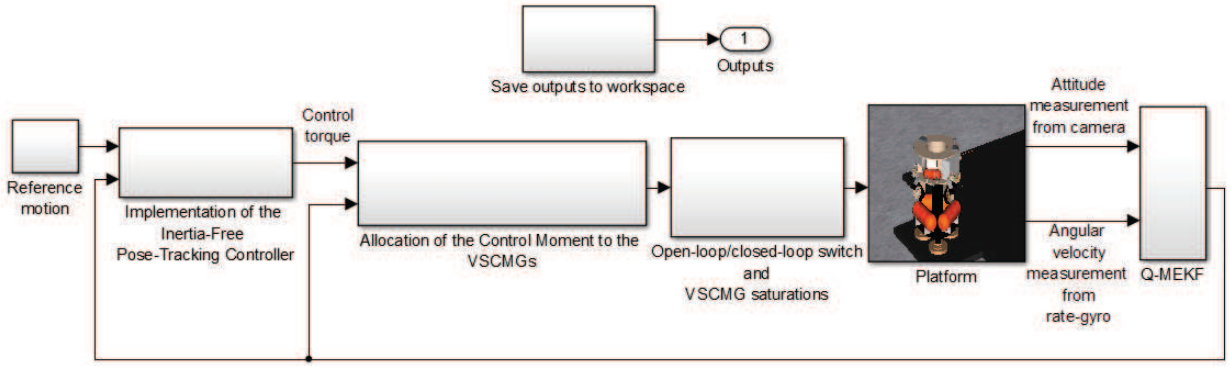


Figure 7: Simulink model implementation of the closed-loop system. The "platform" block includes the whole camera localization system, which is regarded as a measurement source for camera relative poses.

detection with AprilTag [Olson, 2011]. AprilTag is a visual fiducial system for 6DOF camera localization. It will serve as the baseline algorithm for comparison given that it has been comprehensively validated against other fiducial systems including ARToolkitPlus [Wagner et al., 2008] and ARTag [Fiala, 2005], and it was shown to be preferable in terms of localization accuracy, robustness, processing frame-rate, etc [Olson, 2011]. Although ARToolkitPlus has good performance, especially in terms of frame-rate processing, it does not provide high localization accuracy across a large range of distances; it has also poor orientation estimation (in noiseless synthetic scenarios). In contrast, AprilTag provides accurate relative pose across a large range of distances and orientations.

For compatibility with the experimental platform codebase, the C++ version of AprilTag provided by the MIT CSAIL lab was used (`http://people.csail.mit.edu/kaess/apriltags/`). For a fair comparison the payload decoding (tag ID) step was disabled. Tags of comparable sizes were printed. Furthermore, the on-board PC settings, the hardware configuration, the visual environment of the experiment, and the relative poses between the platform and the target, were all fixed during the experiments. The images were captured with a resolution of 640×480 pixels. The time for capturing one frame is 0.005360 ± 0.000291 sec, i.e., the frame rate for only image capturing is 187.0605 ± 9.517 fps. Scoring involved measuring the frame rate of the tag detection and localization procedures, as well as comparing the relative localization against the Vicon$^{\text{TM}}$ ground truth. The experiment involved testing the fiducial marker systems at differing relative distances.

The results of the frame rates are plotted in Figure 9, where the mean and standard deviation versus distance (more than 300 images were taken per distance point) are shown. The results for the relative translation errors and absolute orientation errors versus distance are plotted in Figures 10(a) and 10(b). Both methods share very similar localization accuracy, most likely due to the similarity between the two homography-based localization strategies. However, the value of performing simple blob detection over edge detection followed by edge linking is evident in the achievable frame rates. The proposed method achieves a frame rate that is 5$\sim$6 times faster than that of AprilTag, across the different distances tested. The reason for the improved performance of our method is mainly owing to the fact that the computationally dominant steps, i.e., image rotation, image integration and Dirac convolution, are invariant to the tag distance. AprilTag, on the other hand, is more expensive due to a potentially larger number of linear structures from the environment, all of which must be checked. Examining the AprilTag algorithm, the inclusion of low-pass, Gaussian filtering at a cost of $\mathcal{O}(nmNM)$ where $(n \times m)$ is the size of the filter kernel, already incurs a computational overhead that is higher than the main parts of the Box-LoG detection algorithm. The graph-based gradient clustering and quad (four-sided regions) extraction via depth-first search are more expensive than the previous steps [Olson, 2011], which leads to the reported frame-rate.

Additional advantages of the proposed method against other competing methods is the lower limit of working distance during tracking. The pattern detection step for AprilTag, ARTag, ARToolkitPlus all depend on

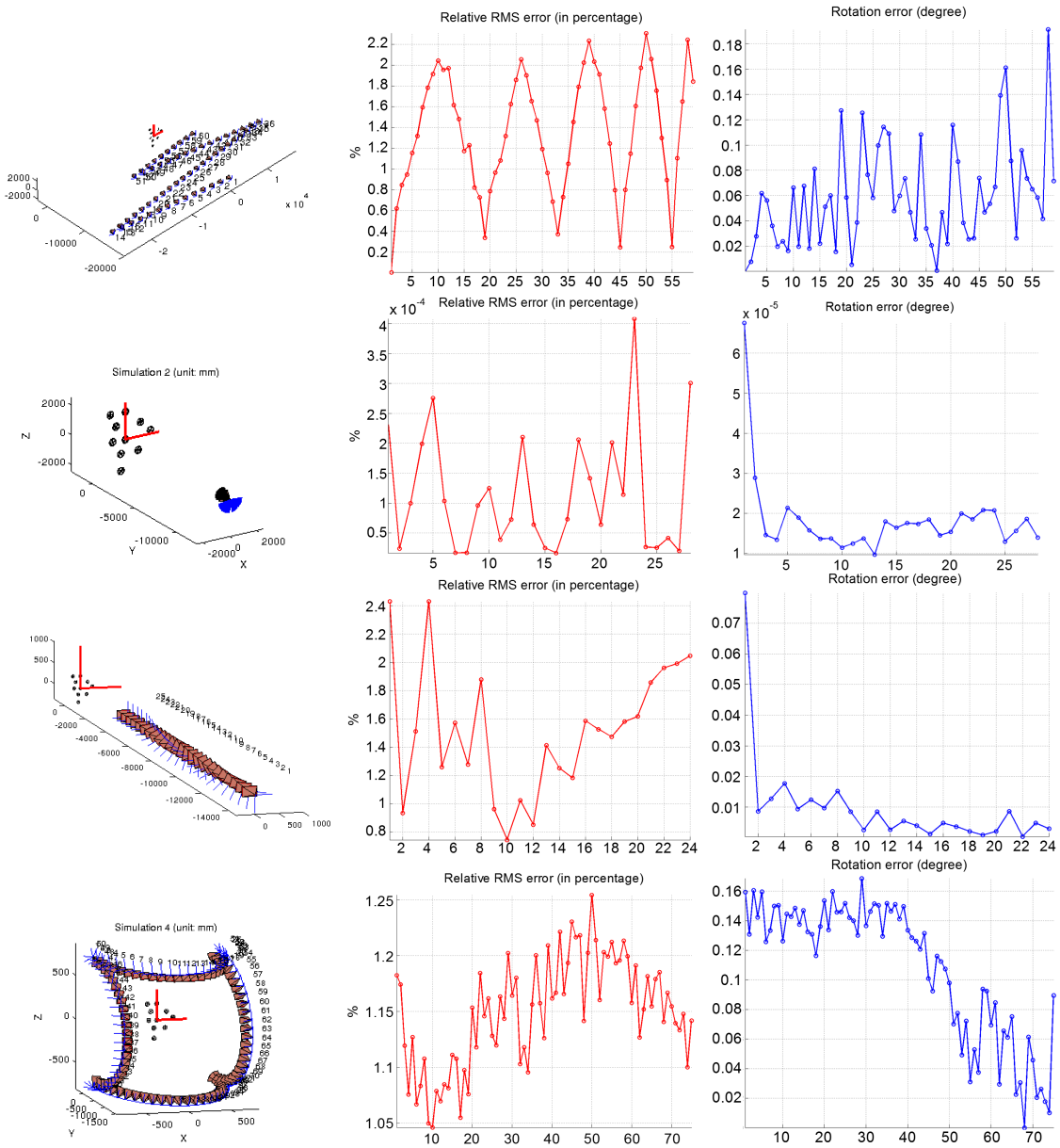


Figure 8: Synthetic experiment results. Column 1: simulated trajectories (units: mm) and camera poses. Column 2: RMS of relative position error versus time for the estimated states. Column 3: orientation error norm versus time for the estimated states.

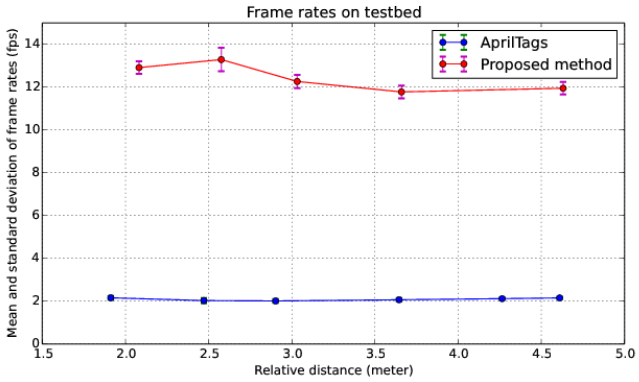
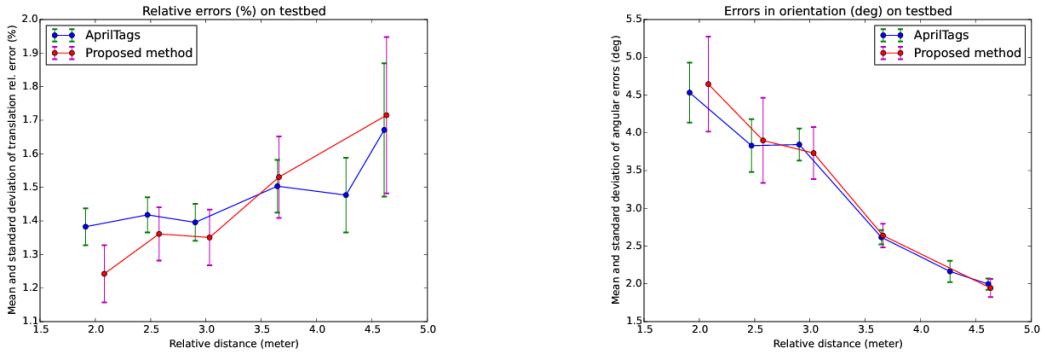


Figure 9: Frame rates of AprilTags algorithm and proposed algorithm on the testbed, under different relative distances. Both the mean values and the standard deviations of frame rates in all instances are shown. The statistics of each instance is computed over a sequence of more than 300 frames.



(a) Relative translation errors of AprilTag and proposed algorithms.

(b) Angular errors of AprilTag and proposed algorithms.

Figure 10: The errors in translation and orientation of the proposed algorithm and AprilTags algorithm respectively from the testbed experiments. Both the mean values and the standard deviations of errors in all instances are shown.

extracting the full boundary of the tag. These methods would fail to pick up the tag as the camera-tag relative distance drops and portions of the tag leave the field-of-view. The proposed blob pattern approach keeps track of the target until there are fewer than three blobs captured. This advantage of our method is particularly important to some proximity operations like docking, where multiple distance scales would have to be traversed during the docking procedure.

Given that the latency of the AprilTag is too high for supporting closed-loop operation, it will not be evaluated in the subsequent sections. The controller is designed in continuous time, and the performance drops significantly if the measurement frequency is lower than 10 Hz. This fact highlights the importance of high computational efficiency in the camera localization algorithm. The proposed tag and detection algorithm is distinguished from the existing tags by its low latency, and high localization accuracy across multiple distance scales.

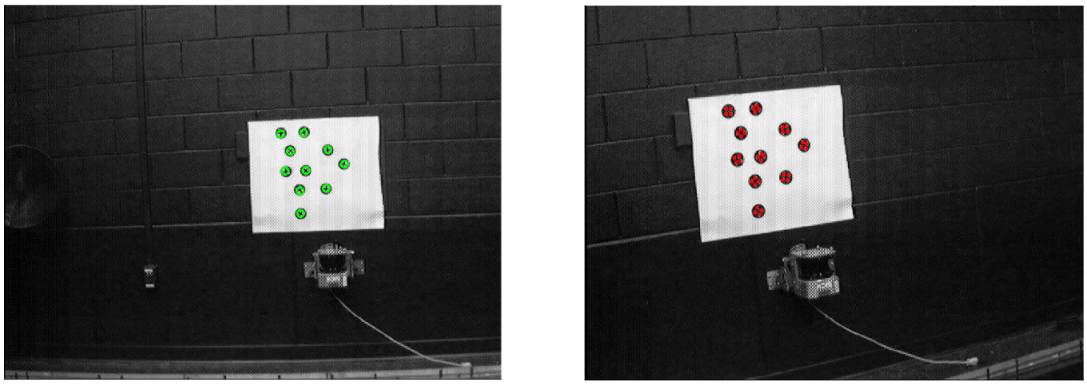


Figure 11: Under different relative distances, blobs of different sizes are automatically selected by the system as the detection target. Left: when the camera is far away from the pattern, the largest size blobs are selected. Right: when the camera is closer to the target, smaller size blobs are selected. The selection is according to the predicted radius (see equation (9)) (the blobs with r_{k+1} closest to 10-pixel is selected). This simple strategy enables multi-scale tracking without much additional computation.

6.3 Open-Loop Relative Pose Estimation Experiments

Prior to testing the closed-loop system, experiments comparable to the simulated system were carried out to test the empirical localization performance of the visual-processing and pose filtering pipeline, as well as the active tag detection system. Two scenarios were tested. In the first scenario, the platform camera follows the (green) trajectory shown in Figure 14(a), which includes translation, rotation, and loss of the target pattern. In the time between poses No. 37 and No. 38 there are three camera image measurements for which the pattern is out of the field of view of the camera, meaning that the pattern is not imaged. In the second experiment, the target pattern is tilted up about 60 degree (y-axis), to test the algorithm's performance under large perspective transformations. The trajectory is recorded as the green line shown in Figure 14(b). For the first half of the trajectory (from frames 1 to 12) the upper stage of the platform is fixed, while for the second half (from frame 12 on) the upper stage of the platform undergoes unknown rotation between camera measurements.

At the beginning of each experiment, the camera is relatively far away from the target. The largest size blobs are automatically selected by the algorithm for detection. As the camera approaches the target, the system switches to detect the blobs of medium scales. The detection results from these two phases are illustrated in Figure 11. When the pattern is acquired, joint data-association and pose estimation is performed. Figure 13 shows a data-association result, in which the detection from the current frame is associated with the previous frame. For both experiments, the final pose estimates are depicted by the camera objects shown in Figures 14(a) and 14(b). Comparing the estimated states from the proposed method to the ground-truth states for both experiments leads to the error plots in Figures 15(a)-16(b).

In the first experiment, the relative errors of the smoothed position estimations are all smaller than 2.8%. The angle deviation between the final estimate rotation matrices and the ground-truth matrices are within 4 deg. For the second experiment the errors of the smoothed pose estimates are below 3.5% (position) and 3.5 deg (orientation). Both experiments confirm the ability of the system to detect and adaptively track the target pattern, as well as to estimate relative pose using the known planar geometry of the pattern elements. In addition, it can be observed that for the position errors, when the camera is closer to the pattern, the relative position errors become smaller. Moreover, compared to the results of the first field experiment under the same relative distance range, the overall errors of these experiments do not increase significantly, which indicates that the perspective image warping due to the placement of the pattern does not affect the performance significantly. Overall, the position and rotation errors are low enough to be used for closed-loop operation with confidence, illustrating the accuracy and robustness of the proposed algorithm.

Figure 12: Detection results of the second field experiment. In this experiment, the medium size blobs are automatically selected as the target markers during the whole experiment.

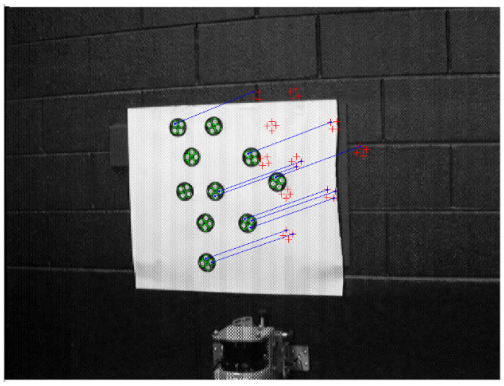
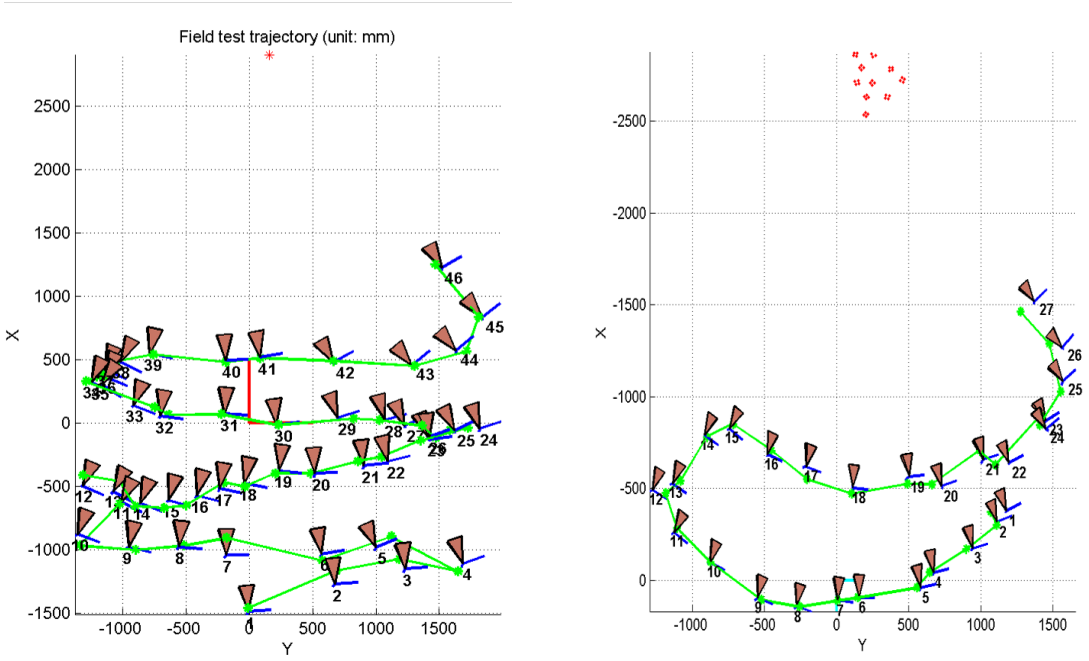


Figure 13: Data association result from two consecutive frames. The red crosses are the target blob positions from the previous frame, while the green dots are the transformed locations of the current frame. The blue line segments stand for the correspondences between the two detected point sets. Note that for clearer illustration, not all the correspondences are plotted, but the data-association results across the whole experiment have been examined to be correct.

6.4 Closed-Loop Relative Attitude Regulation

To validate the performance of the camera localization system in closed-loop, an experiment was run on the ASTROS platform in combination with the inertia-free pose-tracking controller based on dual quaternions from [Filipe and Tsiotras, 2014]. The controller guarantees almost global asymptotic stability[2] of the pose-tracking error without requiring knowledge of the mass and inertia matrix of the platform. Only 3-DOF rotational motion was tested in closed-loop, due to the availability of the actuators at that time. The inputs to the controller are the relative attitude and the relative angular velocity between the platform and the target, with the relative attitude measured using the vision-based localization pipeline. During the experiment, the average time between pose measurements was 0.08127 sec (average pose update rate of 12.30 Hz). The angular velocity of the platform with respect to the inertial frame was measured at 100 Hz with the platform's three-axis rate-gyro (a Humphrey RG02-3227-1 with noise standard deviation of 0.027 deg/s and bias not larger than 2 deg/s). These velocity measurements are filtered by a 4-th order discrete-time Butterworth filter. Since the angular velocity of the target with respect to the inertial frame is not measurable, the controller assumes it is zero and uses the rate-gyro angular velocity measurement as the measurement estimate for the angular velocity between the platform and the target (target motion is

[2]Almost global asymptotic stability is stability over an open and dense set. It the best one can achieve with a continuous controller for orientation, because the group of rotation matrices SO(3) is a compact manifold [Bhat and Bernstein, 2000].



(a) Experiment 1.

(b) Experiment 2.

Figure 14: Estimated trajectories of the the open-loop experiments. Depicted are the ground-truth trajectory of the camera (green line with stars at the actual positions), the target marker positions (red stars), and the final estimated camera poses (camera objects).

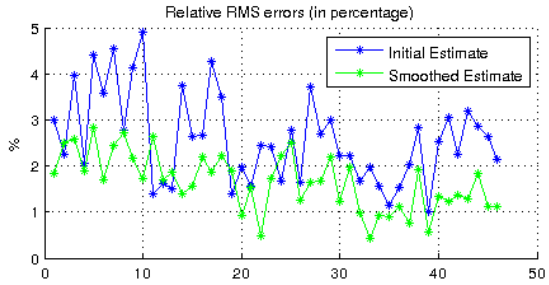
effectively a disturbance).

The measurements of the vector part of the quaternion and angular velocity between the platform and the target were merged in a Quaternion Multiplicative Extended Kalman Filter (Q-MEKF) [Lefferts et al., 1982]. The Q-MEKF is a continuous-discrete Kalman filter (the state and its covariance matrix are propagated continuously between discrete-time measurements). The discrete-time measurements need not be equally spaced in time, making irregular or intermittent measurements easy to handle. Moreover, this structure eases the integration of sensors with different update rates. The states of the Q-MEKF are the quaternion describing the rotation between the platform and the target frame, and the bias of the rate-gyro. The attitude and angular velocity of the platform with respect to the target estimated by the Q-MEKF are fed back to the controller.

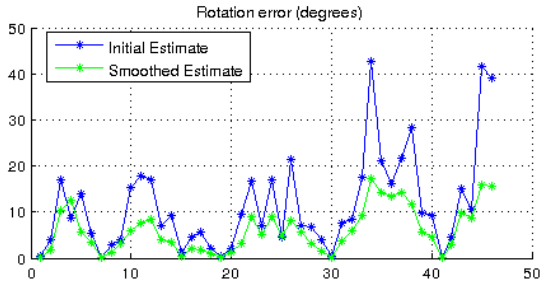
In the sequel, S denotes the platform frame, T denotes the target frame, and D denotes the target's desired frame. During the first 20 sec, no control commands are issued and the Q-MEKF is allowed to converge. Afterwards, the reference attitude is given by $\psi_{\mathrm{D/T}} \equiv -2$ deg, $\theta_{\mathrm{D/T}} \equiv 8$ deg, and $\phi_{\mathrm{D/T}} \equiv -90$ deg, where $\psi_{\mathrm{D/T}}, \theta_{\mathrm{D/T}}, \phi_{\mathrm{D/T}}$ are the three Euler angles in aerospace sequence.

The upper stage is levitated at around 16 sec. At approximately 42 sec after the beginning of the experiment, the target is slowly rotated, leading to a decrease of approximately 3 deg in $\psi_{\mathrm{S/T}}$ and $\theta_{\mathrm{S/T}}$. At approximately 68 sec after the beginning of the experiment, the target is slowly rotated back to its original orientation, leading to an increase of approximately 3 deg in $\psi_{\mathrm{S/T}}$ and $\theta_{\mathrm{S/T}}$. Finally, at approximately 92 sec after the beginning of the experiment, the target is rotated again, leading again to a decrease of approximately 3 deg in $\psi_{\mathrm{S/T}}$ and $\theta_{\mathrm{S/T}}$. The third Euler angle remains approximately constant throughout the experiment.

Figure 17 compares the desired attitude and angular velocity of the S-frame with respect to the T-frame (constant in this experiment) with an estimate of the state of the platform (given by the outputs of the

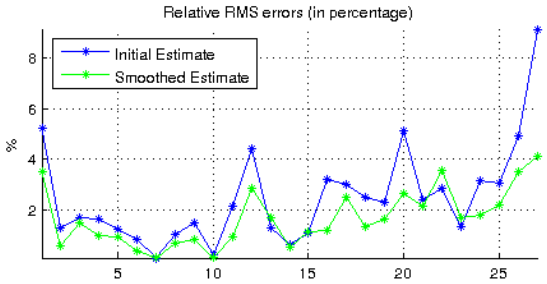


(a) Relative position error.

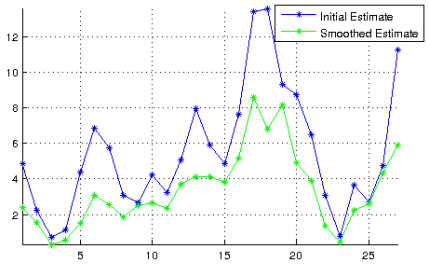


(b) Rotation error.

Figure 15: Experiment 1: Plot of position and orientation error versus frame.



(a) Relative position error.



(b) Rotation error.

Figure 16: Experiment 2: Plot of position and orientation error versus frame.

Q-MEKF). The error between them is presented in Figure 18. After each change in the target orientation, each desired Euler angle is matched within ±2 deg and each desired angular velocity coordinate is matched within ±1 deg/s. This is the same tracking error obtained from previous experiments on the same platform and with the same controller, but with a Crossbow AHRS400CC-100 IMU instead of the camera localization system and rate-gyro, which shows that the error stems primarily from actuator limitations.

7 Conclusions

This paper presents a numerically efficient approach for monocular vision-only relative pose estimation in a cooperative space proximity operations scenario. A cooperative scenario in this context is defined as one where there is a known pattern on the target spacecraft but the target spacecraft motion is unknown. The target pattern, consisting of nested, complementary, contrasting circular blobs in placed asymmetrically, and has been designed specifically to aid detection and localization. The proposed detection strategy employs integer computations where possible, incorporates efficient approximations to costly convolution kernels, and actively employs the closed-loop state estimates to adaptively optimize target detection and localization.

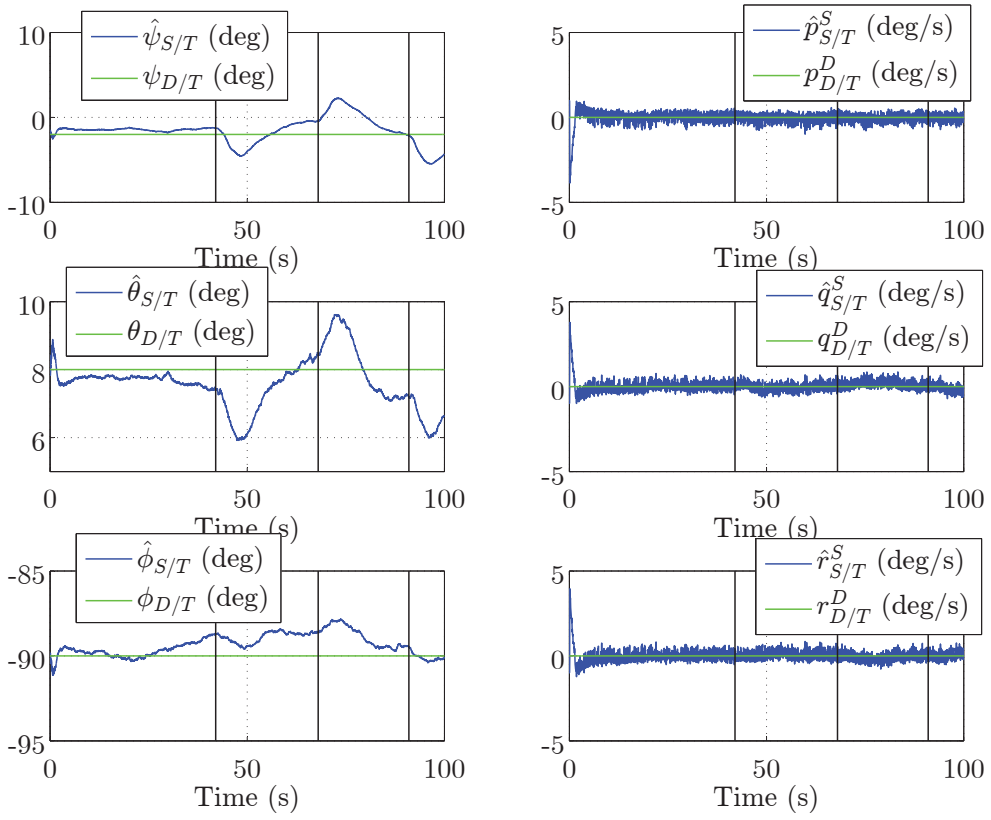


Figure 17: Data from attitude-regulation experiment: desired attitude $(\psi_{\mathrm{D/T}}, \theta_{\mathrm{D/T}}, \phi_{\mathrm{D/T}})$ and angular velocity $(p^D_{\mathrm{D/T}}, q^D_{\mathrm{D/T}}, r^D_{\mathrm{D/T}})$, versus attitude $(\hat{\psi}_{\mathrm{S/T}}, \hat{\theta}_{\mathrm{S/T}}, \hat{\phi}_{\mathrm{S/T}})$ and angular velocity $(\hat{p}^S_{\mathrm{S/T}}, \hat{q}^S_{\mathrm{S/T}}, \hat{r}^S_{\mathrm{S/T}})$ estimated by Q-MEKF. The three vertical lines at 42 sec, 68 sec and 92 sec respectively indicate the starting time of the target movements.

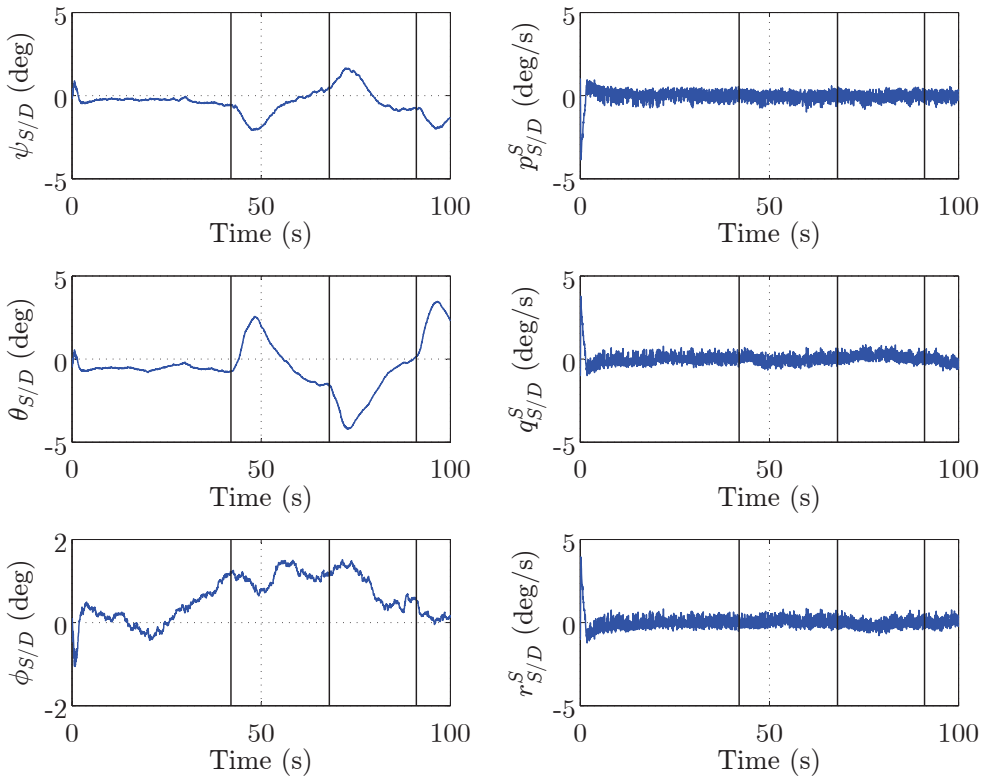


Figure 18: Data from attitude-regulation experiment: attitude $(\psi_{\mathrm{D/T}}, \theta_{\mathrm{D/T}}, \phi_{\mathrm{D/T}})$ and angular velocity $(p^S_{\mathrm{S/D}}, q^S_{\mathrm{S/D}}, r^S_{\mathrm{S/D}})$ regulation error. The three vertical lines at 42 sec, 68 sec and 92 sec respectively indicate the starting time of the target movements.

Marker tracking and frame-to-frame relative pose measurement is done simultaneously by performing point-set registration using a homography-parameterized GMM representation for the detected markers.

The performance of the proposed algorithm has been validated on a 5DOF spacecraft platform capable of simulating realistic spacecraft motion in 1g environment. Open-loop localization test results using the platform demonstrated balanced computational efficiency with good relative pose estimation accuracy. Experiments were also conducted in order to compare the proposed algorithm with AprilTag, a state-of-the-art fiducial-based localization algorithm. These experiments demonstrated a large speedup of the proposed algorithm compared to AprilTag. A closed-loop relative attitude regulation experiment was also performed, which showed that the regulation error rates matched the open-loop estimation error rates, thereby validating the overall controlled active vision system [*why controlled active vision?*]. The algorithm is particularly useful for cooperative navigation between small-size spacecraft, which may have limited on-board power and computation capabilities.

Specifically, compared to existing work, the novel improvements of this work are:

- It provides a solution with low computation and memory complexity and good localization accuracy. The algorithm offers about $\times$ 6 speedup in terms of frame rate, when compared to AprilTag. This improvement is achieved by the novel elements incorporated in the algorithm, including the Box-LoG detector with efficient perspective compensation, the joint data-association and pose estimation, and the feedback framework with homography prediction. The low complexity further helps in improving the update frequency and lowering latency.
- A novel design for cooperative rendezvous using a pattern consisting of a multi-scale blob array, combined with the proposed image processing algorithm, provides robustness against large scale changes for maneuvers in a space environment involving a target with unknown motion status.

Future work includes improving the existing algorithm to accommodate severe illumination changes, typical in space imaging applications. These illumination changes result in large contrast and intensity changes on a single image, which are detrimental to detection accuracy. Another problem left for future investigation is target detection in a cluttered environment. Possible clutter in space includes the Earth in the background, components of the spacecraft (e.g., antennas, solar panels), etc. Background subtraction with more robust target detection algorithms are needed to address the cluttering problem.

References

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

Bhat, S. P. and Bernstein, D. S. (2000). A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon. *Systems & Control Letters*, 39(1):63–70.

Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30.

Cho, D., Tsiotras, P., Zhang, G., and M., H. (2013). Robust feature detection, acquisition and tracking for relative navigation in space with a known target. In *Proc. AIAA Guidance, Navigation, and Control Conference*, Boston, MA.

Cho, D.-M., Jung, D., and Tsiotras, P. (2009). A 5-dof experimental platform for autonomous spacecraft rendezvous and docking. In *Proceedings of AIAA Infotech at Aerospace Conference*, pages 1–20, Seattle, Washington.

Dellaert, F. (2012). Factor graphs and GTSAM: A hands-on introduction. *Georgia Institute of Technology*.

Dellaert, F. and Kaess, M. (2006). Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203.

Delpech, M., Berges, J., Djalal, S., Guidotti, P., and Christy, J. (2012). Preliminary results of the vision based rendezvous and formation flying experiments performed during the PRISMA extended mission. *Advances in the Astronautical Sciences*, 145:1375–1390.

DiMatteo, J., Florakis, D., Weichbrod, A., and Milam, M. (2009). Proximity operations testing with a rotating and translating resident space object. In *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, IL.

Fan, B., Wu, F., and Hu, Z. (2012). Rotationally invariant descriptors using intensity order pooling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2031–2045.

Fehse, W. (2003). *Automated Rendezvous and Docking of Spacecraft*. Cambridge Aerospace Series. Cambridge University Press.

Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 590–596, Washington, DC.

Filipe, N. and Tsiotras, P. (2014). Adaptive position and attitude-tracking controller for satellite proximity operations using dual quaternions. *Journal of Guidance, Control, and Dynamics*. (to appear).

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Flückiger, L. and Utz, H. (2014). Service oriented robotic architecture for space robotics: Design, testing, and lessons learned. *Journal of Field Robotics*, 31(1):176–191.

Gaylor, D. and Lightsey, E. G. (2003). GPS/INS Kalman filter design for spacecraft operating in the proximity of the international space station. In *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, TX.

Hartley, R. and Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, UK.

Hintze, G. C., Cornett, K. G., Rahmatipour, M. H., Heaton, A. F., Newman, L. E., Fleischmann, K. D., and Hamby, B. J. (2007). AVGS, AR&D for satellites, ISS, the Moon, Mars and beyond. In *Proceeding for AIAA Infotech@ Aerospace 2007 Conference and Exhibit, AIAA2007-2883*.

Howard, R. T. and Bryan, T. C. (2007). DART AVGS performance. *NASA Marshall Space Flight Center Technical Report*.

Jasiobedzki, P., Se, S., Bondy, M., and Jakola, R. (2008). Underwater 3D mapping and pose estimation for ROV operations. In *Proc. IEEE conference OCEANS*, pages 1–6, Quebec City, Quebec, Canada.

Jian, B. and Vemuri, B. C. (2005). A robust algorithm for point set registration using mixture of gaussians". In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 1246–1251, Being, China.

Jian, B. and Vemuri, B. C. (2011). Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research*, 31(2):216–235.

Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378.

Karasev, P. A., Serrano, M. M., Vela, P. A., and Tannenbaum, A. (2011). Depth invariant visual servoing. In *IEEE Conference on Decision and Control*, pages 4992–4998, Orlando, FL.

Kasai, T., Oda, M., and Suzuki, T. (1999). Results of the ETS-7 Mission-Rendezvous docking and space robotics experiments. *Proc. 5th Int. Symposium on Articial Intelligence, Robotics and Automation in Space*, 440:299.

Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. 2nd IEEE and ACM International Workshop on Augmented Reality*, page 85, Washington, DC, USA.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519.

Lefferts, E., Markley, F., and Shuster, M. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116.

Lorenz, M., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., and Pollefeys, M. (2012). Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1-2):21–39.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Maire, F. D., Prasser, D., Dunbabin, M., and Dawson, M. (2009). A vision based target detection system for docking of an autonomous underwater vehicle. In *Proc. Australian Conference on Robotics and Automation*, Sydney, Australia.

Negre, A., Pradalier, C., and Dunbabin, M. (2008). Robust vision-based underwater homing using self-similar landmarks. *Journal of Field Robotics*, 25(6-7):360–377.

Neira, J. and Tardos, J. D. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770.

Oda, M. (2001). ETS-VII: Achievements, troubles, and future. In *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: ISAIRAS 2001*.

Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *Proc. IEEE Conference on Robotics and Automation*, pages 3400–3407, Shanghai, China.

Olson, E., Strom, J., Morton, R., Richardson, A., Ranganathan, P., Goeddel, R., Bulic, M., Crossman, J., and Marinier, B. (2012). Progress towards multi-robot reconnaissance and the MAGIC 2010 competition. *Journal of Field Robotics*, 29(5):762–792.

Rekleitis, I., Martin, E., Rouleau, G., L'Archevêque, R., Parsa, K., and Dupuis, E. (2007). Autonomous capture of a tumbling satellite. *Journal of Field Robotics*, 24(4):275–296.

Ruel, S. and Luu, T. (2010). STS-128 on-orbit demonstration of the triDAR targetless rendezvous and docking sensor. In *Proc. IEEE Aerospace Conference*, pages 1–7, Chicago, IL.

Ruel, S., Luu, T., and Berube, A. (2011). Space shuttle testing of the tridar 3d rendezvous and docking sensor. *Journal of Field Robotics*, 29(4):535–553.

Saripalli, S., Montgomery, J. F., and Sukhatme, G. S. (2003). Visually guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–380.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA.

Simard, P. Y., Bottou, L., Haffner, P., and LeCun, Y. (1999). Boxlets: a fast convolution algorithm for signal processing and neural networks. In *Proc. Neural Information Processing Systems*, pages 571–577, Denver, Colorado.

Strasdat, H., Montiel, J. M., and Davison, A. J. (2012). Visual slam: Why filter? *Image and Vision Computing*, 30(2):65–77.

Sun, K., Hess, R., Xu, Z., and Schilling, K. (2014). Real-time robust six degrees of freedom object pose estimation with a time-of-flight camera and a color camera. *to appear Journal of Field Robotics*.

Tsiotras, P. (2014). ASTROS: A 5DOF experimental facility for research in space proximity operations. In *37th AAS Guidance and Control Conference*, Breckenridge, CO. AAS Paper 2014-104.

Tweddle, B. E. (2013). Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations. *PhD, Massachusetts Institute of Technology, Cambridge, MA*.

Velasquez, A. F., Luckett, J., Napolitano, M. R., Marani, G., Evans, T., and Fravolini, M. L. (2009). Experimental evaluation of a machine vision based pose estimation system for autonomous capture of satellites with interface rings. In *Proc. AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois.

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2008). Pose tracking from natural features on mobile phones. In *Proc. IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 125–134, Cambridge, UK.

Wong, V. and Geffard, F. (2010). A combined particle filter and deterministic approach for underwater object localization using markers. In *Proc. IEEE Conference OCEANS*, pages 1–10, Sydney, Australia.

Xu, C., Kuipers, B., and Murarka, A. (2009a). 3D pose estimation for planes. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 673–680. IEEE.

Xu, W., Liang, B., Li, C., Liu, Y., and Wang, X. (2009b). A modelling and simulation system of space robot for capturing non-cooperative target. *Mathematical and Computer Modelling of Dynamical Systems*, 15(4):371–393.

Xu, W., Liang, B., Li, C., and Xu, Y. (2010). Autonomous rendezvous and robotic capturing of non-cooperative target in space. *Robotica*, 28(05):705–718.

Yoon, H. and Tsiotras, P. (2002). Spacecraft adaptive attitude and power tracking with variable speed control moment gyroscopes. *Journal of Guidance, Navigation, and Dynamics*, 25(6):1081–1090.

Zhang, G., Vela, P., Tsiotras, P., and Cho, D. (2014). Efficient closed-loop detection and pose estimation for vision-only relative localization in space with a cooperative target. In *Proc. AIAA Space Conference and Exposition*, San Diego, CA.