Efficient Closed-Loop Detection and Pose Estimation for Vision-Only Relative Localization in Space with A Cooperative Target

Guangcong Zhang[1], Patricio A. Vela[1], Panagiotis Tsiotras[2] and Dae-Min Cho[2*†]

An integrated processing pipeline is presented for relative pose estimation of vision-only cooperative localization between two vehicles with unknown relative motion. The motivating scenario is that of proximity operations between two spacecraft when the target spacecraft has a special target pattern and the chase spacecraft is navigating using only a monocular visual sensor. The only prior information assumed is knowledge of the target pattern, which we propose to consist of nested circular blobs. The algorithm is useful for applications requiring localization accuracy using limited computational resources. It achieves low computational cost with high accuracy and robustness via the following contributions: (1) an adaptive visual pattern detection scheme based on the estimated relative pose, which improves both the efficiency of detection and accuracy of pose estimates; (2) a parametric blob detector called Box-LoG which is computationally efficient; and (3) an algorithm which jointly solves the frame-to-frame data association and relative pose estimation. An incremental smoothing technique temporally smooths the pose estimates. The approach can deal with target re-acquisition after loss of the target pattern from the field of view. The algorithm is tested in both synthetic simulations and on an actual spacecraft simulator platform.

I. INTRODUCTION

I.A. Background

The motivation behind this work arises from the need for a complete processing pipeline achieving efficient, accurate and robust relative pose estimation in computation-limited hardware, e.g. relative poses tracking of a small satellite with respect to a cooperative target. This pose tracking step is significant for autonomous space rendezvous, proximity operations as well as persistent Space Situational Awareness (SSA) applications. Hence, high accuracy is usually required for real-world applications. However, these small spacecraft have limited on-board resources, e.g., power, computation, sensing. Thus, accurate pose tracking under resource limitations is a key unresolved issue. Previously, several techniques have been proposed and tested to solve this problem, which either emphasized the sensory data used (GPS in conjunction with IMU data, LiDAR sensing data, etc.), or used additional aids, such as ground station aided relative navigation. Nevertheless, these techniques suffer from various drawbacks when applied to persistent tracking in space. For example, IMUs experience drift; LiDAR sensors have limited sensing range and are typically active and power hungry; and ground station aided techniques have coarse accuracy, multi-path issues, and possible signal loss.

While passive visual sensors provide a unique set of challenges, recent developments in visual processing indicate that vision-based relative pose estimation may be a feasible, alternative solution for this problem. Since vision sensors have become more accurate, smaller, and of lower power consumption, they are especially suitable for applications in space systems with limited on-board resources for long duration relative pose maneuvering. It should be noted that the same algorithms may aid in cooperative, relative navigation for other robotics applications, e.g., AUV docking, UAV parking, etc.

[*1] Guangcong Zhang and Patricio A. Vela are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA {zhanggc,pvela}@gatech.edu

[†2] Panagiotis Tsiotras (Fellow AIAA) and Dae-Min Cho are with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA {tsiotras,dcho3}@gatech.edu

Problem. In the cooperative satellite proximity operations scenario, the objective is to achieve vision-based, relative navigation about a target satellite via a known pattern placed on the target satellite. Relative navigation should be precise when the known pattern is in view. However, since the pattern may come in and out of the field of view depending on the maneuvers performed by the tracking satellite, the system must be capable of detecting and locking onto the pattern throughout the engagement scenario, as well as recognizing when the pattern has been lost. Due to resource constraints, the processing pipeline should be as computationally efficient as possible, while being robust to uncertainty in the measurements and the uncontrolled relative geometry.

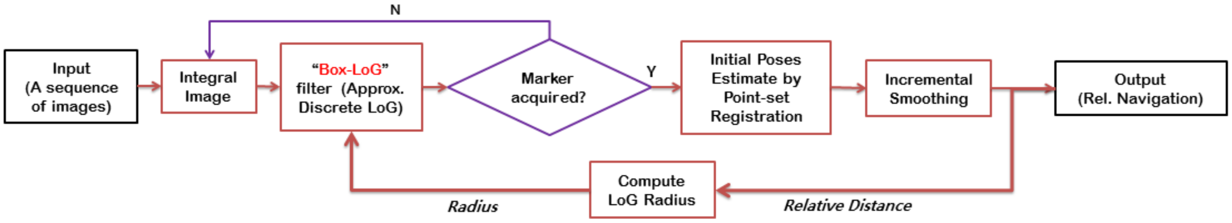


Figure 1. Overall schematic of proposed approach.

Contribution. In addressing the efficiency and accuracy requirements, as well as real-world application issues (e.g., pattern re-acquisition after loss from the camera field of view), this paper describes a complete processing pipeline for relative motion estimation that uses an adaptive detection element for reciprocally using the (prior) pose estimate to improve (current) detection. Furthermore, the paper contributes a new blob detector based on an adaptive, parameterized, piecewise approximation to the Laplacian of Gaussian kernel (LoG) with reduced computational complexity; and a joint frame-to-frame data association and relative pose estimation solver using robust point-set registration with Gaussian mixture models. Finally, an incremental smoothing algorithm temporally smooths the pose estimates, which provides higher accuracy over traditional recursive filtering methods given the same computational overhead [1]. The overall structure of the proposed method is depicted in Figure 1.

The rest of the paper is structured as follows. After a brief summary of related work in the next section, Section III describes the pattern detector and pattern design and Section IV describes the joint pose estimation and data association solution. Section V briefly covers the estimated state smoothing and Section VI demonstrates the accuracy of the algorithm in simulation and using actual data collected using a real 5dof spacecraft simulator platform.

II. Related Work

Relative pose estimation and localization strategies have been developed for various applications including space robotics, AUV (Autonomous underwater vehicle), UAV (Unmanned aerial vehicle), etc. This section attempts to review the most related work across these fields.

Firstly, in terms of the vision system used, the most widely used setup for pose estimate includes stereo and monocular camera systems. Since stereo systems provide directly depth information for mapping, stereo vision based localization and mapping techniques have been proposed for space robotics [2,3] and underwater ROVs [4]. When a stereo system is replaced in favor of a monocular vision system, depth information is lost. Relative pose needs to be estimated by tracking landmarks in the environment frame-by-frame, and posterior optimization is often needed to improve the initial pose estimates. Thus, there are three problems to resolve in monocular vision based pose estimation: (a) feature/landmark detection; (b) data association and pose estimation; and (c) pose filtering.

During the detection phase of an uncooperative target, the landmarks are usually salient visual features detected in the scene. Typical features include geometric structures such corners [5], blobs [6], or more sophisticated features like SIFT [7], SURF [8], and more recently MROGH [9], among others. However, in some applications (e.g., in space), the environment may not have sufficient salient features. Moreover, uncooperative methods have scale ambiguity due to projection transformation. Thus, cooperative methods have been proposed, which assume that some form of a priori knowledge is known. The a priori information is usually the existence of a known pattern on the object. Patterns include special shapes [10], fiducial markers [11], or specially designed patterns such as self-similar landmarks [12], Haar rectangular features [13], 2D bar code style pattern [14], and rings structures [15], etc. Detection of some of these patterns is either computational costly [12], are not robust to large scale changes [10,11,14,15], or they cannot provide accurate 6dof pose estimation [13].

Regarding the data association and pose estimation steps, pose estimation with corresponding features is widely considered a solved problem [16]. It is the data association that remains the key problem. Conventionally, data association is solved by matching the feature descriptors under some mapping criterion [17] or within a robust statistic framework such as RANSAC [18]. However, these techniques rely on the distinctiveness of features. For data association without distinct features, some techniques have been proposed based on point-set matching [11, 19] or image registration [20]. These techniques are especially useful for cooperative cases in which the features from the marker pattern are all similar, such as fiducial dots.

The pose estimates derived from consecutive frames usually suffer from errors. These errors can be reduced through posterior estimation via minimization of the image re-projection errors. Most of the literature for relative pose estimation employs recursive filtering to improve the pose estimates. Popular filtering methods include EKF [21], Particle Filters [22, 23], and their numerous variants [19, 24]. However, recent developments in computer vision demonstrate that (keyframe) batch optimization techniques are more advantageous than filtering-based techniques because the former provides more accuracy per unit of computing time [1]. Smoothing techniques fall under the batch optimization paradigm; incremental smoothing techniques, which are advantageous over traditional fixed-lag/fixed-interval smoothing, have also been developed [25,26], enabling efficient frame-wise smoothing. For space applications only recently has the smoothing framework been used. For example, [11] uses fix-lag smoothing. However, though incremental smoothing has been used for underwater odometry applications [27], it has not been applied to space applications.

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

III. Pattern Detector and Target Pattern Choices

In the scenario considered in this work, pattern detection needs to be invariant to relative orientation about the optical axis, insensitive to the distance from target, and somewhat robust to the perspective distortion caused by angled views of the pattern. Furthermore, the pattern should provide sufficient information to estimate relative pose. The simplest pattern element that fits these constraints, while leading to an equally simple detection algorithm is a blob (a filled circle). This section details a computationally efficient blob detector and describes a pattern, consisting of blob pattern elements, that works at multiple scales (and hence multiple orders of distance).

III.A. Efficient, Adaptive Detection with the Box-LoG Kernel

An optimal parametric detector for blob-like structures across multiple scales is the normalized Laplacian of Gaussian (LoG) detector [6], which applies a normalized and smoothed Laplacian operator $\triangle$ to a 2D field:

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

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 $f(x, y)$, where $x, y \in [-R_{\text{LoG}}, R_{\text{LoG}}] \subset \mathbb{Z}$ and $R_{\text{LoG}} = \lceil 3\sigma \rceil + 1$ to avoid shift artifacts. Appropriately sized blobs in an image I give large magnitude values in the convolved image $I * f$.

In [6], two blob detectors with similar properties were analyzed, the trace of the Hessian (same as Eq. (1)) and the determinant of the Hessian of second order derivatives. The SURF descriptor [8] employs approximations to the determinant of the Hessian by piecewise constant discrete derivatives to achieve efficient operation. However, a piecewise constant trace approximation will have an even lower computational cost with marginal difference in the output.

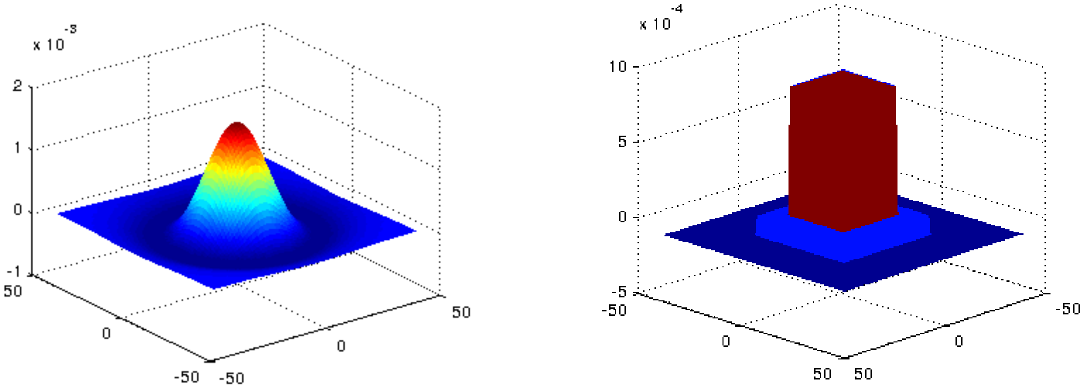


Figure 2. The LoG (left) and Box-LoG (right) kernels

Consider an approximate LoG kernel $g(x,y)$ where $x, y \in [-R_{\text{LoG}}, R_{\text{LoG}}]$ with a three box filters such that:

$$g(x,y) = a_1 H(x,y,R_1) + a_2 H(x,y,R_2) + a_3 H(x,y,R_{\text{LoG}}) \tag{2}$$

where $H(x,y,R)$ is the rectangular function:

$$H(x,y,R) = \begin{cases} 1 & \text{if } x \in [-R,R] \wedge y \in [-R,R], \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The approximate version should satisfy the equations

$$\begin{aligned} \sum_{x,y \in [-R_1,R_1]}\sum f &= \sum_{x,y \in [-R_1,R_1]}\sum g = (a_1 + a_2 + a_3) R_1^2 \\ \sum_{x,y \in [-R_2,R_2]}\sum f &= \sum_{x,y \in [-R_2,R_2]}\sum g = a_1 R_1^2 + (a_2 + a_3) R_2^2 \\ \sum_{x,y \in [-R_{\text{LoG}},R_{\text{LoG}}]}\sum f &= \sum_{x,y \in [-R_{\text{LoG}},R_{\text{LoG}}]}\sum g = a_1 R_1^2 + a_2 R_2^2 + a_3 R_{\text{LoG}}^2 = 0 \end{aligned}$$

Note that the last equation equals to zero because LoG kernel is a zero-mean kernel. The above equations yield a linear system for the coefficients a_1, a_2, a_3, given values of R_1, R_2, and R_{LoG},

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} R_1^2 & R_1^2 & R_1^2 \\ R_1^2 & R_2^2 & R_2^2 \\ R_1^2 & R_2^2 & R_{\text{LoG}}^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum\sum_{[-R_1,R_1]} g \\ \sum\sum_{[-R_2,R_2]} g \\ \sum\sum_{[-R_{\text{LoG}},R_{\text{LoG}}]} g \end{pmatrix}. \tag{4}$$

Since R_{LoG} is a function of σ, the values for R_1 and R_2 need to be specified. Experiments show that when R_1 and R_2 satisfy the relations $(R_1 + R_2)/2 = r$ and $R_2 = 2.5 R_1$, the approximate LoG gives good detection analogous to the continuous LoG. Solving for R_1 and R_2, yields

$$R_1 = \lceil \frac{4}{7} r \rceil \qquad \text{and} \qquad R_2 = 2\,\text{Round}(r) - \lceil \frac{4}{7} r \rceil. \tag{5}$$

Equations (2), (4), (5) fully define the approximate LoG kernel g, henceforth referred to as the *Box-LoG* kernel. An example of a LoG kernel and its Box-LoG approximation are depicted in Figure 2. The Box-LoG kernel is completely specified by the detection radius, similarly to the LoG kernel.

In [8], the determinant of the Hessian blob detector was sped up using integral images, based on the identity [28]

$$J = I * g = \left(\iint I \right) * (g'') \,. \tag{6}$$

The Box-LoG operation benefits from the same property. Since computing the trace is a simpler operation than computing the determinant and, in addition, it does not require the mixed second order derivatives, there will be fewer evaluations of the integral image compared to [8]. The discrete version of the integral image is defined to be:

$$I_{\text{int}}(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y'). \tag{7}$$

The second derivative of Box-LoG consists of a linear combination of eight Dirac delta functions, leading to eight evaluations of I_{int} for the discrete version of J,

$$\begin{aligned} J(x, y) = \sum_{i=0}^{1} \sum_{j=0}^{1} & (-1)^{i+j} (a_1 - a_2) \cdot I_{\text{int}} \left(x + (-1)^i R_1 \, , \, y + (-1)^j R_1 \right) \\ & + (-1)^{i+j} (a_2 - a_3) \cdot I_{\text{int}} \left(x + (-1)^i R_2 \, , \, y + (-1)^j R_2 \right) . \end{aligned}$$

Given I_{int}, computing J involves simple products and sums that take constant runtime. After computing the response image J across 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). Using circular blobs as pattern elements gives rotational invariance about the optical axis and some insensitivity to view-point deviations from the optical axis.

Distance Adaptive Scale Selection. The image formed by a circular marker as seen through a camera depends on the intrinsic camera parameters, the marker's world radius, and the camera to marker distance. For the marker's image radius, there is a direct linear relationship with its world radius and an inverse relationship with the camera-to-marker distance.

During an engagement scenario, this information can be exploited, when it is known. Thus, as part of the processing pipeline, the Box-LoG kernel radius parameter is adapted via feedback of the estimated target position from the previous frame (lower box of Figure 1), as follows

$$r_{k+1} = \frac{\lambda}{\sqrt{\tilde{x}_k^2 + \tilde{y}_k^2 + \tilde{z}_k^2}}, \tag{8}$$

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.

III.B. Landmark Pattern

Given that the pattern consists of circularly symmetric pattern elements (e.g., blobs), whose detection parameters depend on the camera-to-target distance, a pattern is needed that fulfills the remaining constraints. Converting the remaining expectations into a list of features gives:

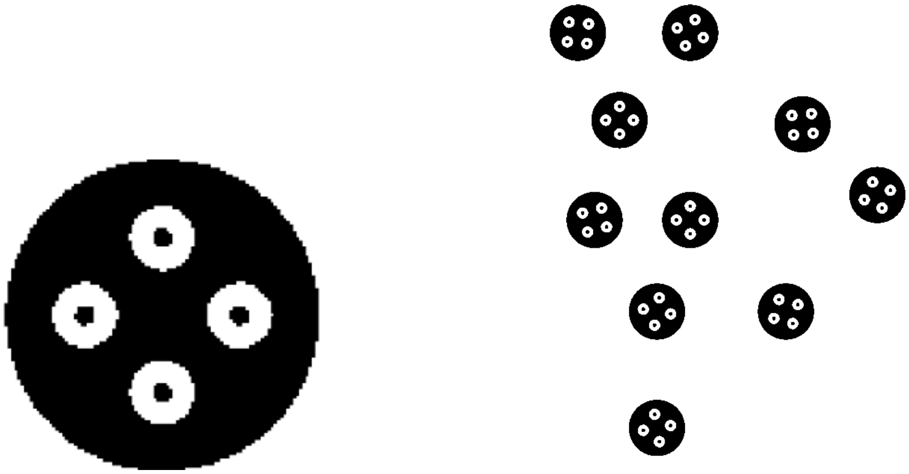
i) pattern elements at multiple scales, for robustness to scale changes;

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

iii) sufficient pattern elements in quantity, for well-posed pose estimation; and

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

A pattern element or marker that achieves the first feature is depicted in Figure 3. The marker consists of nested blobs at different scales and complementary contrasts (dark on light vs. light on dark). The circle radius at one blob size is 4.5 times greater than that of the next nested smaller size. The 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 nor from a neighboring blob scale (the larger blob). When combined with the multi-scale blob detector from Section III.A, the blobs at a given scale can be robustly tracked until the next scale is identified (about two octaves later and with the opposite



(a) Chosen pattern element (b) Candidate pattern

Figure 3. Pattern element and landmark pattern for cooperative tracking.

contrast). Further, the blob markers at the three different scales provide three detection modes determined by the relative distance of the target and the camera.

To fulfill the remaining features, 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 pattern elements leaving the image frame, it is suggested that at least five markers are used and arranged in an asymmetric manner. Moreover, each marker should be least two times its radius away from other markers to avoid false positive detections (in the area between two markers).

The pattern used is shown in Figure 3. It consists of ten pattern elements randomly scattered on a square area, each rotated with a random angle[a]. When the pattern is not being tracked, if more than 4/5 of the pattern is detected, then the pattern is claimed to be re-acquired. During tracking, the Box-LoG detection radius is specified according to the relative distance. 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.

IV. Joint Pattern Tracking and Pose Estimation

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

IV.A. Homography Map

Denote the markers' (homogeneous) locations in the previous image as v_i, the markers' (homogeneous) locations on the current image as u_i, and the 3D positions X_i of the markers on the pattern plane $\pi^\intercal X_i = 0$ with $\pi = (\zeta^\intercal, 1)^\intercal$, $\zeta \in \mathbb{R}^3$, where $i = 1 \dots n_m$ with n_m being the number of markers. For simplicity, let the previous camera pose be the identity pose, i.e., having projection matrix $P_v = [I \,|\, 0]$. If, from the previous to the current frame, the camera rigidly moves by

$$g_v^u = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}, \tag{9}$$

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

then the camera projection on the current frame is

$$P_u = [R \,|\, T]. \tag{10}$$

Since the two views are of points in the same plane, a homography map relates corresponding points between consecutive frames (i.e., it maps v_i to u_i) [16]. The homography map depends on the transformation and the plane normal vector as follows

$$H = (R - T\zeta^{\intercal}). \tag{11}$$

If the point correspondences are known, then the homography, and ultimately the transformation matrix g^u_ν, is computable [16]. Alternatively, if the homography is known, then the points can be placed into correspondence and tracked. The problem arises when neither are known, and the point sets have extra or missing elements (due to false positive or true negatives). To handle these uncertainties, the next section jointly solves the pose estimation and point tracking problems using robust point-set registration.

IV.B. Robust Point-Set Registration

In robust point-set registration, each image point set $\{u_i\}$ and $\{\nu_j\}$ of potentially different cardinality generates a Gaussian Mixture Model (GMM), the first of which is also parameterized by the unknown homography map H. Point-set registration is performed by minimizing the L_2 distance of the GMMs [29,30]. 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.

The GMM generator for a set of points $\mathbf{x} = \{x_i\}_{i=1}^{|\mathbf{x}|}$ is

$$\Phi\,(x\,;\,\mathbf{x}) = \frac{1}{|\mathbf{x}|}\sum_{i=1}^{|\mathbf{x}|}\mathcal{N}\,(x\,;\,x_i,\Sigma), \tag{12}$$

where $|\mathbf{x}|$ is the cardinality of the set $\mathbf{x}$, $\mathcal{N}\,(\cdot\,;\,\cdot,\cdot)$ is the multi-variate normal distribution (the second argument is the mean and the third is the covariance), and Σ is a constant covariance matrix (here, a diagonal matrix with equal variances). The homography map parameterized by the GMM generator is

$$\Phi\,(x\,;\,\mathbf{x},H) = \frac{1}{|\mathbf{x}|}\sum_{i=1}^{|\mathbf{x}|}\mathcal{N}\,(x\,;\,Ax_i + b, A\Sigma A^{\intercal}), \tag{13}$$

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

$$\operatorname{dist}(\Phi\,(\cdot\,;\,\mathbf{u},H)\,,\Phi\,(\cdot\,;\,\mathbf{v})) \triangleq \int (\Phi\,(x\,;\,\mathbf{u},H) - \Phi\,(x\,;\,\mathbf{v}))^2\,\mathrm{d}x$$

The multi-variate Gaussian distribution obeys the identity

$$\int \mathcal{N}\,(x\,;\,\mu_1,\Sigma_1)\mathcal{N}\,(x\,;\,\mu_2,\Sigma_2)\,\mathrm{d}x = \mathcal{N}\,(0\,;\,\mu_1-\mu_2,\Sigma_1+\Sigma_2).$$

As a result, $\operatorname{dist}(\Phi\,(\cdot\,;\,\mathbf{u},H)\,,\Phi\,(\cdot\,;\,\mathbf{v}))$ can be computed in closed-form as follows

$$\begin{aligned}\operatorname{dist}(\Phi\,(\cdot\,;\,\mathbf{u},H)\,,\Phi\,(\cdot\,;\,\mathbf{v})) = {} & \frac{1}{|\mathbf{u}|^2}\sum_{i=1}^{|\mathbf{u}|}\sum_{j=1}^{|\mathbf{u}|}\mathcal{N}\,(0\,;\,A(u_i-u_j), 2A\Sigma A^{\intercal}) \\ & + \frac{1}{|\mathbf{v}|^2}\sum_{i=1}^{|\mathbf{u}|}\sum_{j=1}^{|\mathbf{v}|}\mathcal{N}\,(0\,;\,\nu_i-\nu_j, 2\Sigma) \\ & - 2\frac{1}{|\mathbf{u}||\mathbf{v}|}\sum_{i=1}^{|\mathbf{u}|}\sum_{j=1}^{|\mathbf{v}|}\mathcal{N}\,(0\,;\,H(u_i)-\nu_j, A\Sigma A^{\intercal}+\Sigma).\end{aligned}$$

Minimization of the distance is performed iteratively through gradient descent. The first two terms are constant, having no effect on the optimization, and thus they do not factor into the iterations. The $SE(3)$ transformation is then recovered by minimizing $\text{dist}(\Phi\left(\cdot\,;\,\mathbf{u}, H\right), \Phi\left(\cdot\,;\,\mathbf{v}\right))$ over H,

$$\begin{aligned} H &= \arg\min_H \text{dist}(\Phi\left(\cdot\,;\,\mathbf{u}, H\right), \Phi\left(\cdot\,;\,\mathbf{v}\right)) \\ &= \arg\max_H \frac{1}{|\mathbf{u}||\mathbf{v}|} \sum_{i=1}^{|\mathbf{u}|} \sum_{j=1}^{|\mathbf{v}|} \mathcal{N}\left(0\,;\, H(u_i) - \nu_j, A\Sigma A^\intercal + \Sigma\right). \end{aligned}$$

After finding H, two points u_i and ν_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 threshold. The transformation g_ν^u in (9) is determined using the matrix entries of H given ζ [16].

V. Smoothing the Pose Estimates

While the pose estimates are optimized for the current observations conditioned on the previous observations (Section IV.B), they are not optimized temporally over all observations (e.g., they are not filtered). For vision-based measurements, temporal filtering is performed by minimizing the image re-projection errors given the set of pose estimates and homographic mappings to date.

Denote by $\mathcal{G} \triangleq \{g_\tau\}$ the set of camera poses at time instants τ, by $\mathcal{L} \triangleq \{l_j\}$ the constant set of target pattern landmarks where j ranges over all possible landmark indices, by $\mathcal{Z} \triangleq \{\zeta_\tau\}$ the collection of measurements where ζ_τ consists of the points $\{u_i\}$ that were imaged at time τ (when indicating the time instant, we will write $u_{i,\tau}$, In addition, there is a time-dependent association function $\alpha_\tau(\cdot)$ that matches a measurement index to a landmark index (this function is instantiated when the pattern is detected and maintained during marker tracking). Define the measurement function $h(g, l)$ to be the perspective camera projection mapping 3D points to 2D image coordinates. Given a measurement and landmark association, the image re-projection error for measurement index i at time τ is:

$$\varepsilon_{i,\tau} = h(g_\tau, l_{\alpha_\tau(i)}) - u_{i,\tau}. \tag{14}$$

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

$$P\left(u_{i,\tau}|g_\tau, l_{\alpha_\tau(i)}\right) \propto \exp\left(-\frac{1}{2}\|\varepsilon_{i,\tau}\|^2_{\Sigma_k}\right). \tag{15}$$

Define $\Theta \triangleq (\mathcal{G}, \mathcal{L})$ as the collection of the unknown camera poses and landmarks, and model the system with a factor graph. In our case, no odometry information is available, therefore the factors that encode the prediction model do not exist. Using the factorization property of factor graphs, the joint probability of all random variables is

$$P(\Theta) \propto \left(\prod_\tau \varphi_\tau(\theta_\tau)\right)\left(\prod_{\tau,j} \psi_{\tau,j}(\theta_\tau, \theta_j)\right), \tag{16}$$

where τ ranges over the variables in $\mathcal{G}$, j ranges over the variables in $\mathcal{L}$, the potentials $\varphi_\tau(\theta_\tau)$ encode a prior estimate at $\theta_\tau \in \Theta$, and the pairwise potentials $\psi_{\tau,j}(\theta_\tau, \theta_j)$ encode information between two factors (here, a camera pose and a landmark). Using this information, the potentials are

$$\varphi_\tau(\theta_\tau) \propto P(g_\tau) \tag{17}$$

$$\psi_{\tau,j}(\theta_\tau, \theta_j) \propto P(u_{\alpha_\tau^{-1}(j),\tau}|g_\tau, l_j). \tag{18}$$

For the second set of potentials, the potential (and hence factor graph edge) does not exist when the inverse is not defined for a given (τ, j) (i.e., the landmark was not seen). The maximum a posteriori (MAP) estimate is

$$\begin{aligned} \hat{\Theta} &= \arg\max_\Theta P(X, L|Z) = \arg\max_\Theta P(X, L, Z), \\ &= \arg\min_\Theta(-\log P(\Theta)). \end{aligned}$$

Since the information is arriving in time, the incremental smoothing method [25, 26] is used for optimizing the pose estimates. We use the GTSAM library, written by the authors of the above references, to implement the incremental smoothing step.

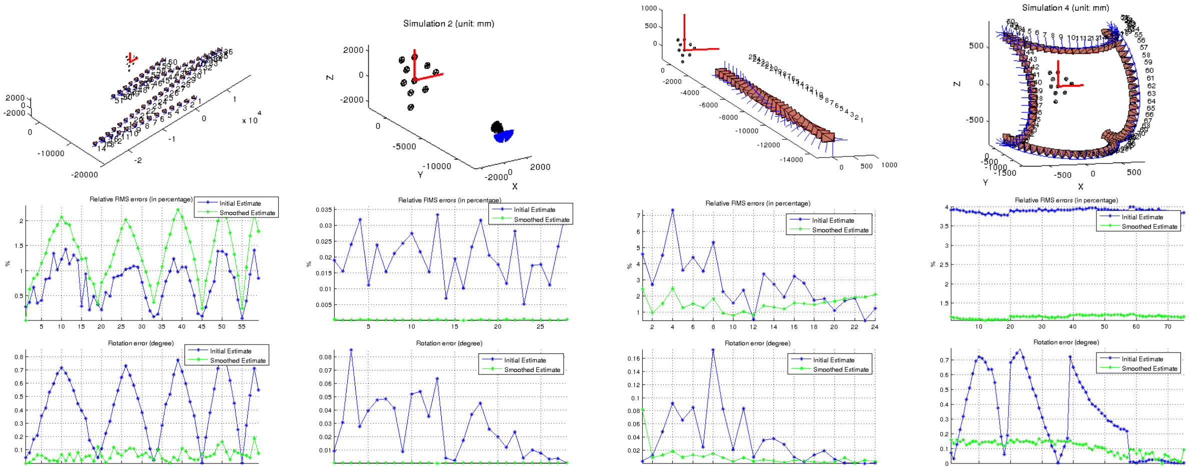


Figure 4. Synthetic experiment results. Row 1: simulated trajectories (in unit mm) and camera poses. Row 2: RMS of relative position error versus time for the estimated and smoothed states. Row 3: orientation error norm versus time for the estimated and smoothed states.

VI. Experimental Results

This section evaluates the processing pipeline of Figure 1, as described in the previous sections, on both synthetic and actual relative movement scenarios. Accuracy is evaluated for both position and orientation separately. Position accuracy is measured as a percentage using the relative norm of the relative position error. 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 is $100\|\tilde{X} - X\|_2 / \|X - X_T\|_2$. The orientation accuracy is given by the error of estimated camera orientation computed via the norm of the $SO(3)$ logarithm converted to degrees. To this end, let $\tilde{R}$ be the estimated orientation and let R be the ground-truth orientation. Then the error is

$$\frac{180}{\pi} \left\| \left(\log_{SO(3)}(\tilde{R}^{\intercal} R) \right)^{\vee} \right\| \tag{19}$$

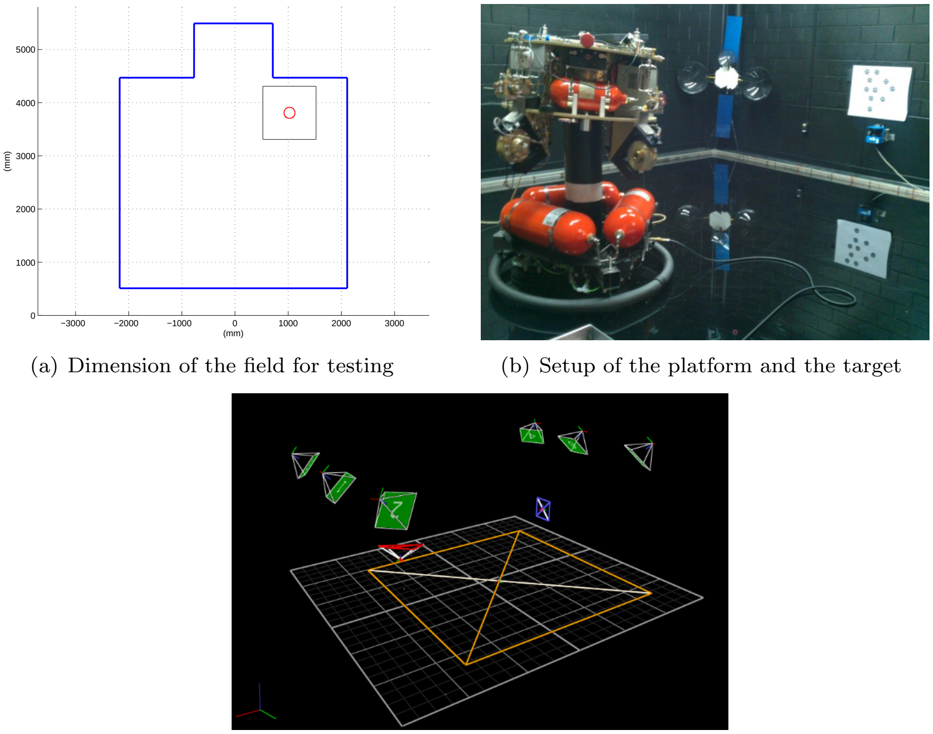
where the "unhat" operation $(\cdot)^{\vee}$ maps a 3×3 skew-symmetric matrix to a vector.

VI.A. Synthetic Image Experiment

The synthetic experiment is conducted with Matlab. In the experiment, a 3D virtual reality environment with the designed target is first simulated. Then a simulated camera moves along a designated trajectory and captures images of the target according to a pinhole camera projection model. The focal length of the simulated camera is $1,388$ and the resolution is 1082×722. The algorithm is then tested on these synthetic images and the results are evaluated comparing to the ground-truth. In this experiment, we assume no distortion in the camera projection and no noise in the camera movements.

Four trajectories were simulated. The trajectories and the (measurement) camera poses for each simulation are shown in the first row of Figure 4. Each simulation 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 observability properties that influence the relative position and orientation estimates in different ways. Most motions also involve large perspective changes over the course of the trajectory which also tests the (adaptive) pattern detection algorithm.

In Figure 4, the second and third rows 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 smaller than 0.2°. The results demonstrate that the proposed algorithm can detect the pattern, estimate the relative poses accurately, and adapt the detection scale accordingly.



(a) Dimension of the field for testing

(b) Setup of the platform and the target

(c) Vicon^{TM} setup

Figure 5. Experimental testbed. The Vicon^{TM} setup depicts the marker cameras (green), platform frame (red), target pattern frame (blue) and field floor (orange)

VI.B. Field Experiments

The proposed algorithm was also tested on a 5dof spacecraft simulator testbed (Autonomous Spacecraft Testing of Robotic Operations in Space - ASTROS), whose components are depicted in Figure 5. The spacecraft (seen in Figure 5(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 x 14 ft, and the air pads activated, the spacecraft experiences almost friction-free conditions. The main structure of the ASTROS is the upper stage, whose main operational characteristics can be found in Ref. [31]. 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 vision capturing, a CCD camera (TMS-730p by Pulnix) mounted on the test bed is used. The camera digitizer resolution is 640×480. A six camera Vicon^{TM} system is used to capture the ground-truth pose of the upper stage of the platform (and related to the camera frame by a rigid transformation estimated as part of system calibration) and the target pattern pose.

VI.B.1. Experiment 1

The camera follows the (green) trajectory shown in Figure 6. The platform trajectory 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.

VI.B.2. Experiment 2

In the second experiment, the target pattern is tilted up about 60 degree (y-axis). This setup is to test the algorithm's performance under large perspective transformations. Moreover, in the first half of the trajectory (from frames 1 to 12) the upper stage of the platform is fixed, while in the second half (from frame 12 on) the upper stage of the platform undergoes unknown rotation between camera measurements. The trajectory is recorded as the green line shown in Figure 7.

VI.B.3. Experimental Results

For both experiments, the smoothed pose estimates are depicted by the camera objects shown in Figures 6 and 7. Comparing the estimated and smoothed states to the ground-truth states for both experiments leads to the error plots in Figures 8 - 11. With regards to 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 18°. For the second experiment the errors of the smoothed pose estimates are below 5% (position) and 10° (orientation). Both experiments confirm the ability of the system to detect and adaptively track the target pattern, as well as estimate relative pose using the known planar geometry of the pattern elements.

VII. Conclusions

This paper presents an efficient processing pipeline for vision-only relative pose estimation in a cooperative scenario, where the target pattern is specifically crafted to have specific invariance and the detection strategy is designed to have specific robustness properties. In particular, the planar, asymmetric pattern consists of pattern elements with nested, complementary contrasting circular blobs. A radially adaptive Box-LoG detector is proposed which approximates the LoG kernel and has low computational cost. The radius of the detector is adapted with the feedback of the poses estimation to avoid multi-scaled detection and increase the detection accuracy. Marker tracking and frame-to-frame poses measurement is done simultaneously by performing point-set registration using a homography parameterized GMM representation for the detected markers. The final estimated states are incrementally smoothed. Experimental and simulation results show that the proposed approach is able to estimate relative poses efficiently under scale changes and with high accuracy and robustness. The next step is to perform closed-loop relative navigation.

Acknowledgement: This work has been supported by AFRL research award FA9453-13-C-0201.

References

[1] Strasdat, H., Montiel, J. M., and Davison, A. J., "Visual SLAM: Why filter?" *Image and Vision Computing*, Vol. 30, No. 2, 2012, pp. 65–77.

[2] Xu, W., Liang, B., Li, C., and Xu, Y., "Autonomous rendezvous and robotic capturing of non-cooperative target in space," *Robotica*, Vol. 28, No. 05, 2010, pp. 705–718.

[3] Xu, W., Liang, B., Li, C., Liu, Y., and Wang, X., "A modelling and simulation system of space robot for capturing non-cooperative target," *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 15, No. 4, 2009, pp. 371–393.

[4] Jasiobedzki, P., Se, S., Bondy, M., and Jakola, R., "Underwater 3D mapping and pose estimation for ROV operations," *Proc. IEEE conference, OCEANS 2008*, 2008, pp. 1–6.

[5] Shi, J. and Tomasi, C., "Good features to track," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[6] Lindeberg, T., "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision*, Vol. 30, No. 2, 1998, pp. 79 – 116.

[7] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91–110.

[8] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, Vol. 110, No. 3, 2008, pp. 346–359.

[9] Fan, B., Wu, F., and Hu, Z., "Rotationally invariant descriptors using intensity order pooling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 10, 2012, pp. 2031–2045.

[10] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Visually guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 3, 2003, pp. 371–380.

[11] Cho, D., Tsiotras, P., Zhang, G., and Holzinger, M., "Robust Feature Detection, Acquisition and Tracking for Relative Navigation in Space with a Known Target," *Proc. AIAA Guidance, Navigation, and Control Conference*, 2013.

[12] Negre, A., Pradalier, C., and Dunbabin, M., "Robust vision-based underwater homing using self-similar landmarks," *Journal of Field Robotics*, Vol. 25, No. 6-7, 2008, pp. 360–377.

[13] Maire, F. D., Prasser, D., Dunbabin, M., and Dawson, M., "A vision based target detection system for docking of an autonomous underwater vehicle," *Proc. Australasion Conferenece on Robotics and Automation*, 2009.

[14] Olson, E., "AprilTag: A robust and flexible visual fiducial system," *Proc. IEEE International Conferenece on Robotics and Automation*, IEEE, 2011, pp. 3400–3407.

[15] Velasquez, A. F., Luckett, J., Napolitano, M. R., Marani, G., Evans, T., and Fravolini, M. L., "Experimental Evaluation

of a Machine Vision Based Pose Estimation System for Autonomous Capture of Satellites with Interface Rings," *Proc. AIAA Guidance, Navigation, and Control Conference*, 2009.

[16] Hartley, R. and Zisserman, A., *Multiple view geometry in computer vision*, Cambridge Univ Press, 2000.

[17] Neira, J. and Tardós, J. D., "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 6, 2001, pp. 890–897.

[18] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395.

[19] Wong, V. and Geffard, F., "A combined particle filter and deterministic approach for underwater object localization using markers," *Proc. IEEE Conference OCEANS 2010*, 2010, pp. 1–10.

[20] Karasev, P. A., Serrano, M. M., Vela, P. A., and Tannenbaum, A., "Depth invariant visual servoing," *IEEE Conference on Decision and Control*, 2011, pp. 4992–4998.

[21] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, 2007, pp. 1052–1067.

[22] Montemerlo, M. and Thrun, S., *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, Springer, 2007.

[23] Augenstein, S. and Rock, S. M., "Simultaneous estimation of target pose and 3-D shape using the FastSLAM algorithm," *Proc. AIAA Guidance, Navigation, and Control Conference*, 2009.

[24] Augenstein, S. and Rock, S. M., "Improved frame-to-frame pose tracking during vision-only SLAM/SFM with a tumbling target," *Proc. IEEE International Conference on Robotics and Automation*, 2011, pp. 3131–3138.

[25] Kaess, M., Ranganathan, A., and Dellaert, F., "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, Vol. 24, No. 6, 2008, pp. 1365–1378.

[26] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F., "iSAM2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotics Research*, Vol. 31, No. 2, 2012, pp. 216–235.

[27] Kim, A. and Eustice, R. M., "Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency," *IEEE Transactions on Robotics*, 2013, pp. 719 – 733.

[28] Simard, P. Y., Bottou, L., Haffner, P., and LeCun, Y., "Boxlets: a fast convolution algorithm for signal processing and neural networks," *Proc. Neural Information Processing Systems*, 1999, pp. 571–577.

[29] Jian, B. and Vemuri, B. C., "A robust algorithm for point set registration using mixture of Gaussians," *Proc. IEEE International Conference on Computer Vision*, Vol. 2, IEEE, 2005, pp. 1246–1251.

[30] Jian, B. and Vemuri, B. C., "Robust point set registration using Gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 8, 2011, pp. 1633–1645.

[31] Cho, D., Jung, D., and Tsiotras, P., "A 5-DOF Experimental Platform for Spacecraft Rendezvous and Docking," *AIAA Infotech at Aerospace Conference*, April 6–9 2009, Seattle, WA.

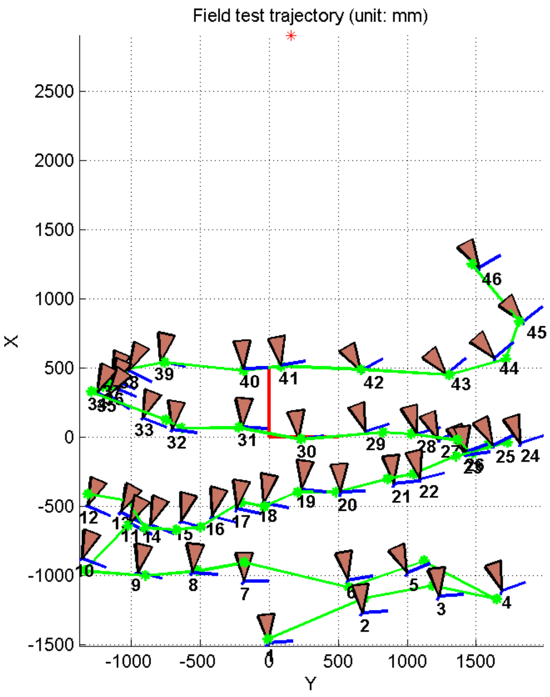


Figure 6. Experiment 1. The ground-truth trajectory (in green) of the camera, the target position (in red star) and the final estimated camera poses (depicted by the camera objects)

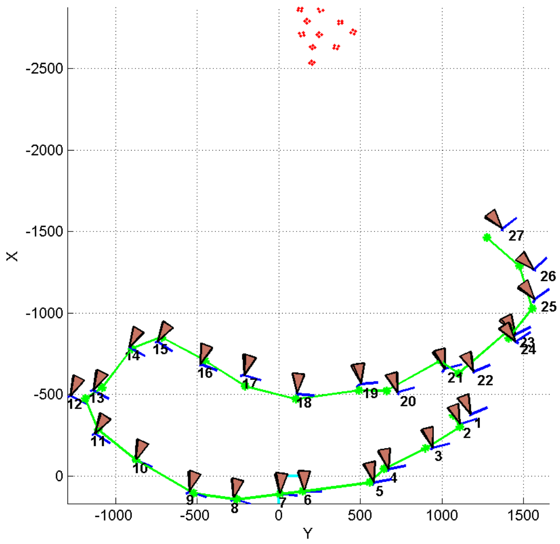


Figure 7. Experiment 2. The ground-truth trajectory (in green) of the camera, the target position (in red star) and the final estimated camera poses (depicted by the camera objects)

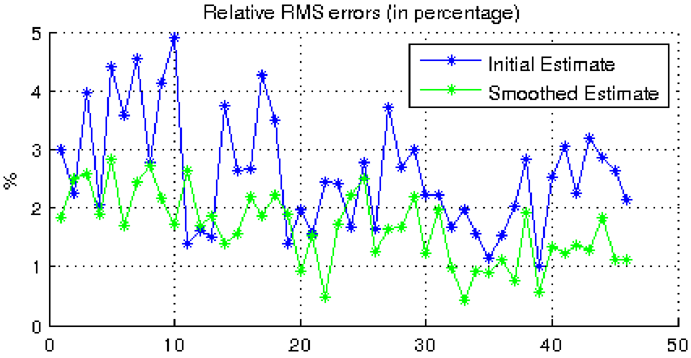


Figure 8. Experiment 1. Relative errors of the estimated positions.

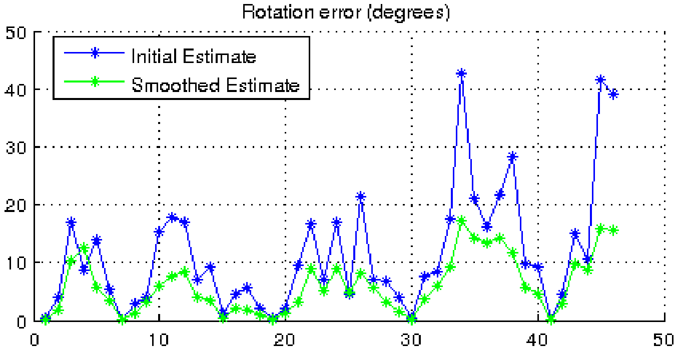


Figure 9. Experiment 1. Rotation errors.

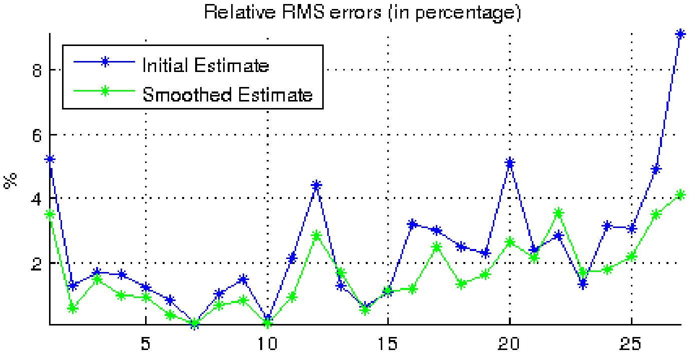


Figure 10. Experiment 2. Relative errors of the estimated positions.

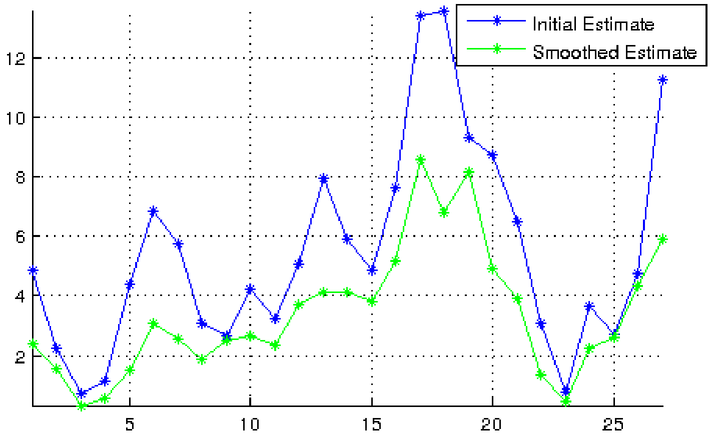


Figure 11. Experiment 2. Rotation errors.