sensors

MDPI

Article

Tracking a Non-Cooperative Target Using Real-Time Stereovision-Based Control: An Experimental Study

Tomer Shtark † and Pini Gurfil *,†

Department of Aerospace Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel; tomershtark@gmail.com

* Correspondence: pgurfil@technion.ac.il; Tel.: +972-4-829-3020

† These authors contributed equally to this work.

Academic Editor: Joonki Paik

Received: 14 February 2017; Accepted: 28 March 2017; Published: 31 March 2017

Abstract: Tracking a non-cooperative target is a challenge, because in unfamiliar environments most targets are unknown and unspecified. Stereovision is suited to deal with this issue, because it allows to passively scan large areas and estimate the relative position, velocity and shape of objects. This research is an experimental effort aimed at developing, implementing and evaluating a real-time non-cooperative target tracking methods using stereovision measurements only. A computer-vision feature detection and matching algorithm was developed in order to identify and locate the target in the captured images. Three different filters were designed for estimating the relative position and velocity, and their performance was compared. A line-of-sight control algorithm was used for the purpose of keeping the target within the field-of-view. Extensive analytical and numerical investigations were conducted on the multi-view stereo projection equations and their solutions, which were used to initialize the different filters. This research shows, using an experimental and numerical evaluation, the benefits of using the unscented Kalman filter and the total least squares technique in the stereovision-based tracking problem. These findings offer a general and more accurate method for solving the static and dynamic stereovision triangulation problems and the concomitant line-of-sight control.

Keywords: tracking; stereovision; real-time control

1. Introduction

1.1. Cooperative vs. Non-Cooperative Targets

An important problem in the field of computer vision is estimating the relative pose. Numerous past studies addressed this problem, dealing with objects on which some information is a priori available; these objects are referred to as *cooperative targets*.

To estimate the relative pose with respect to cooperative targets, Fasano, Grassi and Accardo [1] used light emitting diodes, while Woffinden and Geller [2] used known shapes as features, placed at known positions on the target. Terui, Kamimura and Nishida [3] used a known 3D model of the object, a model matching technique, and 3D feature points obtained from stereo matching.

The problem of estimating the position, velocity, attitude and structure of a non-cooperative target significantly surpasses the cooperative problem in complexity. Segal, Carmi and Gurfil [4,5] used two cameras for finding and matching a priori unknown feature points on a target. These feature points were fed to an Iterated Extended Kalman Filter (IEKF) [6], which was based on the dynamics of spacecraft relative motion.

Also, the target's inertia tensor was estimated using a maximum a posteriori identification scheme. Lichter and Dubowsky [7,8] used several vision sensors, which were distributed fairly uniformly

about the target, in order to estimate its properties. Sogo, Ishiguro and Trivedi [9] used multiple omnidirectional vision sensors and a background subtraction technique in order to measure the azimuth angle of the target from each vision sensor. Jigalin and Gurfil [10] compared the Unscented Kalman Filter (UKF) [11–13] and the IEKF, and studied the effects of increasing the number of cameras.

The problem of controlling robot motion using visual data as feedback, is commonly referred to as visual servoing. Cai, Huang, Zhang and Wang [14] presented a novel dynamic template matching that may be used for a monocular visual servoing scheme of a tethered space robot in real time using one camera. The proposed matching method detects specific regions on satellites with high accuracy. It also includes experiments for verifying the proposed matching method, showing promising results. Chen, Huang, Cai, Meng and Liu [15] proposed a novel non-cooperative target localization algorithm for identification of a satellite's brackets, which is a proper grasping position for a tethered space robot. Reference [15] also includes experiments on standard natural images and bracket images, which demonstrated its robustness to changes in illumination. Huang, Chen, Zhang, Meng and Liu [16] proposed a control method of visual servoing based on the detection of a margin line on the satellite brackets, using only one camera. Reference [16] used a gradient-based edge line detection method for finding the satellites brackets and acquiring its relative position and attitude.

1.2. Computer Vision Algorithms

An important computer vision technique needed for relative pose estimation is feature detection and matching. This method searches images for visual information, which can be utilized to identify objects in other images, while being partially or fully invariant to scaling, translation, orientation, affine distortion, illumination changes, clutter and partial occlusion. There are numerous feature point detection, description and matching methods [17].

Lowe [18] proposed the Scale Invariant Feature Transform (SIFT), which is invariant to orientation, translation, and scale. It describes each feature point using 128-element long vectors, which are used to determine matching between features in different images. These vectors are also referred to as *descriptors*. Bay, Ess, Tuytelaars and Van Gool [19] proposed the Speeded-Up Robust Features (SURF) algorithm, which is faster and more efficient than SIFT. The task of finding corresponding feature points between two images is transformed into matching descriptors from two groups. It is done by using the nearest neighbour ratio matching strategy [17–19], and a technique called Random Sample Consensus (RANSAC) [17,20,21], which is used to eliminate outliers.

Another important problem in computer vision is recognition of an unknown target in a given image. An unknown target is an object with a high uncertainty concerning its shape and location in the image. Jigalin and Gurfil [10] presented a method of segmentation and identification of an unknown target out of a group of potential targets in an image. They assumed some rough characteristics of the target's shape and location, and that the target can be distinguished from the background. A series of morphological image processing operators were applied on the image in order to distinguish all the potential targets from their background. Then, the morphological properties of the target candidates were compared using a cost function, and a "best" target candidate was selected.

In the current research, the target recognition algorithm developed in Ref. [10] is expanded by checking if the "best" target meets certain criteria in order to be considered as a legitimate target. With the modified algorithm, the assumption that the target is always in the image is redundant. Moreover, this research is aimed at developing and implementing the mentioned computer vision algorithms using a specialized testbed. For a scalability analysis of the effects of larger distances on the computer vision algorithms, the reader is referred to Ref. [22].

1.3. Relative State Estimators

After the target was detected, it is desired to estimate its relative position and velocity with respect to the chaser. To this end, an estimator is required. In this research, three estimators, EKF, UKF and Converted Measurement Kalman Filter (CMKF), are compared and examined using experimental data.

The CMKF is less known compared to the EKF and UKF. The main idea in CMKF is to convert the measurement equation into a linear equation by an algebraic manipulation. The noise models in the converted equations are not necessarily identical to the noise model in the original equations, and, as a result, a biased estimation might occur. The CMKF was first developed by Lerro and Bar-Shalom [23] for tracking systems that measure the position of targets in polar coordinates. They also added bias compensation components, referred to as "Debiased CMKF". Later on, Suchomski [24] proposed a three-dimensional version of this filter. A second version of the CMKF was proposed by Xiaoquan, Yiyu and Bar-Shalom [25], which was referred to as "Unbiased CMKF". It included different bias compensation components. Later on, it was also developed using a different bias compensation technique [26].

An important issue with implications on the estimation process is the Depth Error Equation (DEE), which is an analytical equation that yields an approximation of the Line-of-Sight (LOS) vector's depth component. The DEE can be expanded to all of the LOS components, and is referred to as Relative Position Error Equations (RPEE). The RPEE yields an analytical approximation of the expected errors of the relative position components. Gallup, Frahm, Mordohai and Pollefeys [27] investigated the DEE with changing baseline and resolution in order to compute a depth map over a certain volume with a constant depth error. Oh et al. [28] evaluated the DEE via experiments.

A direct solution of the stereo measurement model is referred to as Converted Measurements (CMS). The CMS are used for initialization of the estimators, and as a reliability check for each measurement. That is, if the CMS and the state estimation are too dissimilar, than the measurement's validity should be questioned. In the case of stereovision with two horizontally positioned and vertically aligned cameras, the measurement model is comprised of four algebraic, coupled, overdetermined and non-linear equations. By rearranging these equation, the model's non-linearity can be eliminated at the cost of certain numerical issues. In this case, the CMS can be attained by various methods, such Least Squares (LS) [6] and Total Least Squares (TSL) [29].

1.4. Research Contributions

The current research is an experimental effort, in which non-cooperative target tracking methods using stereovision measurements are developed. Real-time closed-loop line-of-sight control based on stereovision measurements only is developed and evaluated. In particular, our main contribution is the development of a stereovision tracking algorithm of non-cooperative targets suitable for real-time implementation. Moreover, we examine different estimation schemes, and evaluate their real-time performance. An emphasis is given on devising efficient methods for initializing the relative position estimators based on stereovision measurements.

The remainder of this paper is organized as follows. Section 2 introduces the computer vision models and algorithms used throughout this study. In Section 3, the estimation models are introduced, including the CMKF, which is a novel estimator. Section 4 describes the control algorithm, which is implemented in the experiments. In Section 5 a numerical analysis of the problem described in Section 3.4 is conducted. Section 6 describes the Distributed Space Systems Laboratory (DSSL) hardware, the experimental system, and three experiments, which implement the algorithms developed in this research. Section 7 contains the conclusions of this work.

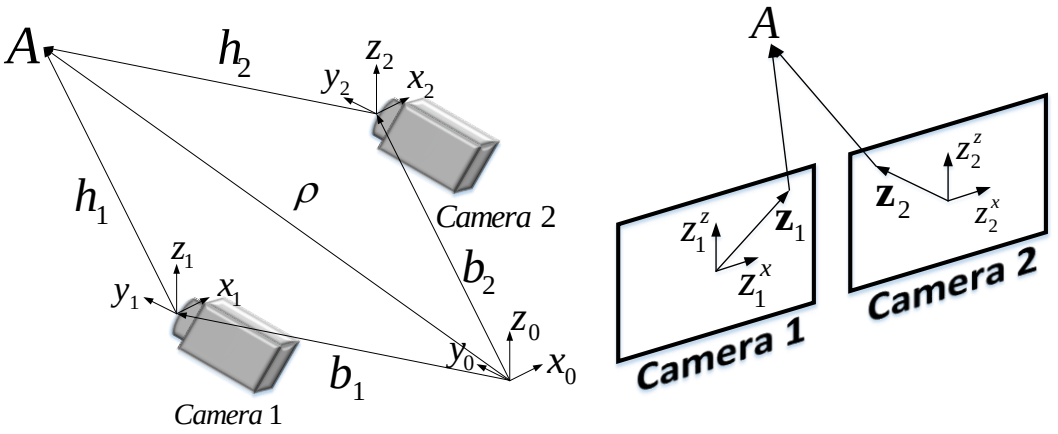
Notice that Sections 2, 3 and 5 address a 3D problem, while Section 4 addresses a 2D problem. The reason for that is that the estimation algorithms are designed with the intention to be general, while the control algorithm has to be adjusted to the 2D setup of the robots in the DSSL. As a result, the control algorithm uses only the relevant 2D components of the estimated 3D quantities.

2. Computer Vision Models and Algorithms

2.1. Pinhole Model

Consider N cameras, which are only separated by translation, and have no relative rotation, positioned at N different fixed points in a three-dimensional space, facing the same direction, and sharing a common image plane. A reference coordinate system is denoted by $\{x_0, y_0, z_0\}$. The camera-fixed coordinates are denoted by $\{x_i, y_i, z_i\}$. The reference coordinate system and the camera-fixed coordinate systems are rotating together and are facing the same direction at all times, so no transformation is needed between them. The vectors $\vec{b}_i \ \ (i = 1, ..., N)$ connect the reference coordinate system with the N camera-fixed coordinates.

The vector $\vec{\rho}$ connects $\{x_0, y_0, z_0\}$ with an arbitrary point in space, A. Assuming A appears in the Fields of View (FOV) of all the cameras, the vectors $\vec{h}_i \ \ (i = 1, ..., N)$ connect each camera's respective coordinate origin with point A. Without loss of generality, it is assumed that all the cameras are facing the y direction of their respective coordinates. This geometry is depicted in Figure 1a.



(a) A reference frame and two cameras sharing a common image plane, observing the same point in space.

(b) Projection of a point on two image planes of horizontally positioned and vertically aligned cameras.

Figure 1. The geometry of stereovision.

Each camera's image plane has a planar coordinate system $\{z_i^x, z_i^z\}$, whose origin is positioned in the center of each image plane, as depicted in Figure 1b. The vectors $\vec{z}_i$ connect the center of the image plane of camera i to the projection of point A on the image plane. They have a horizontal component z_i^x, a vertical component z_i^z, and are measured in pixels.

It is important to distinguish between the coordinate frames $\{x_i, y_i, z_i\}$ and $\{z_i^x, z_i^z\}$. The former is a three-dimensional coordinate frame, located in a three-dimensional space, in which distances are dimensional; the latter is a two-dimensional coordinate frame, located on the image plane, in which distances are measured in pixels. The pinhole camera model [17] yields the following mathematical relation for each camera:

$$\begin{pmatrix} z_i^x \\ z_i^z \end{pmatrix} = \frac{1}{h_i^y} \begin{pmatrix} f_i^x h_i^x \\ f_i^z h_i^z \end{pmatrix} \tag{1}$$

where f_i^x and f_i^z are the focal lengths of each camera (measured in pixels) in the x and z directions, respectively, and h_i^x, h_i^y, h_i^z are the components of vector $\vec{h}_i$ in the reference coordinate system. From Figure 1a, it can be seen that

$$\vec{\rho} = \vec{b}_i + \vec{h}_i \tag{2}$$

Therefore, the *non-linear projection equations* are

$$\begin{pmatrix} z_i^x \\ z_i^z \end{pmatrix} = \frac{1}{\rho_y - b_i^y} \begin{pmatrix} f_i^x(\rho_x - b_i^x) \\ f_i^z(\rho_z - b_i^z) \end{pmatrix} \tag{3}$$

These equations can also be rearranged in the following manner, which is referred to as the *linear projection equations*:

$$\begin{pmatrix} 1 & -z_i^x/f_i^x & 0 \\ 0 & -z_i^z/f_i^z & 1 \end{pmatrix} \begin{pmatrix} \rho_x \\ \rho_y \\ \rho_z \end{pmatrix} = \begin{pmatrix} 1 & -z_i^x/f_i^x & 0 \\ 0 & -z_i^z/f_i^z & 1 \end{pmatrix} \begin{pmatrix} b_i^x \\ b_i^y \\ b_i^z \end{pmatrix} \tag{4}$$

When all the cameras share the same plane, meaning $b_y^i = 0$, Equations (3) and (4) yield

$$\begin{pmatrix} z_i^x \\ z_i^z \end{pmatrix} = \frac{1}{\rho_y} \begin{pmatrix} f_i^x(\rho_x - b_i^x) \\ f_i^z(\rho_z - b_i^z) \end{pmatrix} \qquad i = 1, ..., N \tag{5}$$

$$\begin{pmatrix} 1 & -z_i^x/f_i^x & 0 \\ 0 & -z_i^z/f_i^z & 1 \end{pmatrix} \begin{pmatrix} \rho_x \\ \rho_y \\ \rho_z \end{pmatrix} = \begin{pmatrix} b_i^x \\ b_i^z \end{pmatrix} \qquad i = 1, ..., N \tag{6}$$

2.2. Image Resizing

Consider an image with the dimensions X_1 and Y_1 and focal lengths f_{base}^x, f_{base}^z in the x and y directions, respectively. Assume that it is desired to resize that image to the dimensions X_2 and Y_2. To that end, R_F is defined as the *Resize Factor*. For simplicity, we assume that the image is resized while keeping the aspect ratio constant. The relation between X_1, X_2, Y_1, Y_2 and R_F is

$$X_2 = R_F\, X_1 \quad , \quad Y_2 = R_F\, Y_1 \tag{7}$$

When images are resized, their focal length should be corrected correspondingly,

$$\begin{aligned} f^x &= R_F f_{base}^x \\ f^z &= R_F f_{base}^z \end{aligned} \tag{8}$$

where f^x and f^y are the focal lengths of the resized image.

2.3. Relative Position Measurement Error Approximation

The error of the measured relative position $\vec{\rho}_m$ for a simple case of two horizontally positioned and vertically aligned cameras, is approximated in the following manner. It is assumed that the focal lengths of the cameras are equal,

$$f_1^x = f_1^z = f_2^x = f_2^z = f \tag{9}$$

In Figure 2, depicting the geometry of aligned cameras, it can be seen that $z_1^x > 0$, $z_2^x < 0$, $b_1^x < 0$, $b_2^x > 0$. Therefore,

$$\begin{aligned} b &= b_2^x - b_1^x > 0 \\ d &= z_1^x - z_2^x > 0 \end{aligned} \tag{10}$$

where b is the distance between the cameras, also referred to as *baseline*, and d is the *disparity* [17,21]. The linear projection Equations (6) for this case become

$$\begin{bmatrix} 1 & -\frac{1}{f} z_1^x & 0 \end{bmatrix} \begin{bmatrix} \rho_x & \rho_y & \rho_z \end{bmatrix}^T = b_1^x \tag{11}$$

$$\begin{bmatrix} 1 & -\frac{1}{f} z_2^x & 0 \end{bmatrix} \begin{bmatrix} \rho_x & \rho_y & \rho_z \end{bmatrix}^T = b_2^x \tag{12}$$

$$\begin{bmatrix} 0 & -\frac{1}{f}z_1^z & 1 \end{bmatrix} \begin{bmatrix} \rho_x & \rho_y & \rho_z \end{bmatrix}^T = 0 \tag{13}$$

$$\begin{bmatrix} 0 & -\frac{1}{f}z_2^z & 1 \end{bmatrix} \begin{bmatrix} \rho_x & \rho_y & \rho_z \end{bmatrix}^T = 0 \tag{14}$$

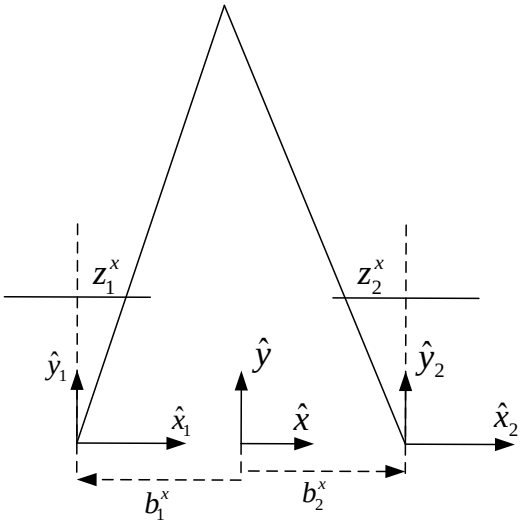


Figure 2. A two dimensional geometry of two aligned cameras and a point's projection on their respective image planes.

Equations (11) and (12) are referred to as the *Horizontal Equations*, while Equations (13) and (14) are referred to as the *Vertical Equations*. These equations can be solved using the Combined Vertical Equations (CVE) method. The CVE approach, developed in this research, addresses the specific problem of stereovision with two horizontally positioned and vertically aligned cameras. According to this approach, it is noticed that the vertical equations in both cameras are identical up to measurement errors. Therefore, to render the linear system determined, the vertical equations are averaged into one equation. Consequently, a solution can be achieved by simply inverting the matrix in the left-hand side of Horizontal Equations, which becomes a 3×3 matrix. The solutions according to the CVE method for ρ_x, ρ_y, ρ_z are given by

$$\begin{aligned} \rho_x &= \frac{z_1^x b_2^x - z_2^x b_1^x}{z_1^x - z_2^x} \\ \rho_y &= \frac{f(b_2^x - b_1^x)}{(z_1^x - z_2^x)} \\ \rho_z &= \frac{(z_1^z + z_2^z)(b_2^x - b_1^x)}{2(z_1^x - z_2^x)} = \frac{z_1^z + z_2^z}{2f}\rho_y \end{aligned} \tag{15}$$

The horizontal error, depth error and vertical error, denoted by $\Delta\rho_x$, $\Delta\rho_y$, $\Delta\rho_z$, respectively are defined as

$$\begin{gathered} \Delta\rho_i = \sqrt{\left(\frac{\partial\rho_i}{\partial z_1^x}\Delta z_1^x\right)^2 + \left(\frac{\partial\rho_i}{\partial z_2^x}\Delta z_2^x\right)^2 + \left(\frac{\partial\rho_i}{\partial z_1^z}\Delta z_1^z\right)^2 + \left(\frac{\partial\rho_i}{\partial z_2^z}\Delta z_2^z\right)^2} \\ i = \{x, y, z\} \end{gathered} \tag{16}$$

where Δz_1^x, Δz_2^x, Δz_1^z, Δz_2^z are the errors in z_1^x, z_2^x, z_1^z, z_2^z, respectively. Furthermore, it is assumed that

$$\Delta z \approx \Delta z_1^x \approx \Delta z_2^x \approx \Delta z_1^z \approx \Delta z_2^z \tag{17}$$

By combining Equations (15)–(17), the following expressions are obtained:

$$\Delta\rho_x = \frac{\sqrt{(z_1^x)^2+(z_2^x)^2}(b_2^x-b_1^x)}{(z_1^x-z_2^x)^2}\Delta z = \frac{\rho_y}{bf}\sqrt{(\rho_x-b_1^x)^2+(\rho_x-b_2^x)^2}\Delta z$$
$$\Delta\rho_y = \frac{\sqrt{2}f(b_1^x-b_2^x)}{(z_1^x-z_2^x)^2}\Delta z = \frac{\sqrt{2}fb}{d^2}\Delta z = \frac{\rho_y^2}{bf}\sqrt{2}\Delta z \quad (18)$$
$$\Delta\rho_z = \sqrt{\frac{(b_2^x-b_1^x)^2}{2(z_1^x-z_2^x)^2}+\frac{(z_1^z+z_2^z)^2(b_2^x-b_1^x)^2}{2(z_1^x-z_2^x)^4}}\Delta z = \sqrt{\frac{\rho_y^2}{2f^2}+\frac{2\rho_y^2\rho_z^2}{b^2f^2}}\Delta z = \frac{\rho_y}{bf}\left(\frac{b^2}{2}+2\rho_z^2\right)^{\frac{1}{2}}\Delta z$$

Finally, the relative position vector is approximated as

$$\vec{\rho} = \begin{bmatrix}\rho_x\\ \rho_y\\ \rho_z\end{bmatrix} \approx \begin{bmatrix}\rho_m^x\\ \rho_m^y\\ \rho_m^z\end{bmatrix} + \begin{bmatrix}\Delta\rho_x\\ \Delta\rho_y\\ \Delta\rho_z\end{bmatrix} = \vec{\rho}_m + \Delta\vec{\rho} \quad (19)$$

where $\vec{\rho}_m$ is the *converted measurement*, calculated using Equations (11)–(14), and $\Delta\vec{\rho}$ is the additive error vector.

2.4. Computer Vision Algorithms

To utilize the stereovision camera as a sensor, the target has to be identified in the scene, following the extraction of its position in the image plane in all cameras. To that end, a methodology which relies on feature detection and matching, image segmentation and target recognition algorithms is used.

2.4.1. Feature Detection and Matching

Feature detection and matching [17] is used for matching the target in all cameras, by detecting feature points in the images and assigning to each point a descriptor with encoded visual information. Consequently, it is possible to find the same point in different images by matching the descriptors numerically.

Over the years, different kinds of feature point detection methods were developed. The most well known is SIFT [18], and there are others such as SURF [19], Maximally Stable Extremal Regions (MSER) [30], Features from Accelerated Segment Test (FAST) [31] and Oriented fast and Rotated Brief (ORB) [32]. Each method relies on a different geometric concept, and has different properties, such as computational effort and reliability.

In this research, SURF was used. The computational effort of SURF is significantly smaller than SIFT, which was used by Jigalin and Gurfil [10]. The feature matching methods used herein are:

1. Nearest Neighbour—The descriptors of the feature points of all the cameras are matched by comparing them using Euclidian distances. The reliability of the matches increases by eliminating ambiguous matches, which are defined as matches that have a low ratio between their distance and the distance of the second-closest neighbor (see [18]).
2. Fundamental Matrix Estimation Using RANSAC [21]—This method eliminates the outliers of the Nearest Neighbour method by applying geometric constraints.
3. Slope Cut Off—This method eliminates more outliers by enforcing geometric consistency. It places the stereovision images side by side and stretches lines between all the matched points. For every object, all the lines should have a similar slope, so the lines should not cut each other. The matches of lines that do not adhere to these rules are declared as outliers and rejected.

2.4.2. Image Segmentation

The following image segmentation method [10] is used for locating all the target candidates in the image plane. A target candidate's projection on the image plane is assumed to be a notable blob. With this assumption, the images from all the cameras are subjected to the following procedures

1. The images are processed by a Sobel edge detector [17], which returns binary images of the edges in the scene.
2. The binary images are filtered in order to clean noisy pixels.
3. The images are processed through an image dilation operator, which expands the outlines of the objects in the scene. This operation closes all the gaps in the objects' outlines.
4. After the gaps in the objects' outlines are closed, the objects, which are closed outlined blobs, are filled.
5. Finally, the images are processed by an erosion operator, which diminishes all the blobs in the images. This operation leaves only the most significant blobs.

All the remaining separate blobs are considered as target candidates.

2.4.3. Target Recognition

The next step is to determine which of the obtained target candidates is the real one. In order to do so, the following method is used. The method starts by calculating the area, solidity and distance of each target candidate's region. *Area* is the number of pixels in each region. *Solidity* is the area fraction of the region with respect to its convex hull ($0 < Solidity < 1$). *Distance* is the mean distance of SURF feature points in each region.

The distance value for each feature point is simply the y component of the LOS vector $\vec{\rho}$ to each feature point. These vectors are obtained by solving Equation (6) for each feature point. An important issue that has to be dealt with, is the method which is used to solve Equation (6). Among other methods, it can be solved using LS or TLS, which were mentioned above. A comparison is given in Section 5.

The properties (*Area*, *Solidity* and *Distance*) of each target candidate has to be bounded within certain upper and lower bounds. Each target candidate, whose properties are not limited to these bounds, is neglected. Among the target candidates, which meet the previous criteria, the true target is selected using the relationship

$$Target\ Region = \max_{i=1}^{N_c} \left(\frac{\sqrt{Area_i} \cdot Solidity_i}{Distance_i} \right) \tag{20}$$

where N_c is the number of the target candidates. It can be seen that this formula gives a preference to larger, closer and more convex objects using a non-dimensional value.

3. Estimation of the Relative State

3.1. Process Model

We define $\mathcal{I}$ as a cartesian right-hand inertial reference frame. The state vector, $\vec{x}$, contains the relative position and velocity between the chaser and the target in the inertial coordinate frame $\mathcal{I}$,

$$\vec{x} = \begin{bmatrix} \vec{\rho}^T & \dot{\vec{\rho}}^T \end{bmatrix}^T \tag{21}$$

To write a process model, it is assumed that the target's mass, inertia, applied forces and torques are unknown; therefore, an exact dynamical model, which describes the relative state between the chaser and the target, is quite complicated to derive. Instead, a white-noise acceleration model [6] can

be implemented. This model is less accurate, but also does not require much information regarding the target. The continuous process model is written as

$$\dot{\vec{x}} = A\vec{x} + \vec{v} \quad , \quad A = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{22}$$

where $\vec{v}(t) \in \mathbb{R}^{6\times1}$ is a white Gaussian process noise vector defined as

$$\vec{v} = \begin{bmatrix} 0 & 0 & 0 & \nu_{\dot{x}} & \nu_{\dot{y}} & \nu_{\dot{z}} \end{bmatrix}^T \tag{23}$$

satisfying

$$E[\vec{v}(t)] = 0 \quad , \quad E[\vec{v}(t)\vec{v}^T(t+\tau)] = \vec{q_\nu}(t)\delta(t-\tau) = Q_c \tag{24}$$

with Q_c being the power spectral density, and

$$\vec{q_\nu} = \begin{bmatrix} 0 & 0 & 0 & \sigma_x^2 & \sigma_y^2 & \sigma_z^2 \end{bmatrix}^T \tag{25}$$

The next step is to write the discrete-time process model with a sampling period Δt,

$$\vec{x}_{k+1} = \Phi\vec{x}_k + \vec{v}_k \tag{26}$$

where Φ is the discrete state transition matrix,

$$\Phi = e^{A\Delta t} = \begin{bmatrix} I_3 & I_3\Delta t \\ 0_{3\times3} & I_3 \end{bmatrix} \tag{27}$$

The discrete process noise vector $\vec{v}_k$ satisfies

$$E(\vec{v}_k) = 0 \tag{28a}$$

$$E(\vec{v}_k\vec{v}_k^T) = Q = \int_0^{\Delta t} e^{A\tau} Q_c e^{A^T\tau} d\tau$$

$$Q = \begin{bmatrix} \frac{\sigma_x^2\Delta t^3}{3} & 0 & 0 & \frac{\sigma_x^2\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\sigma_y^2\Delta t^3}{3} & 0 & 0 & \frac{\sigma_y^2\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\sigma_z^2\Delta t^3}{3} & 0 & 0 & \frac{\sigma_z^2\Delta t^2}{2} \\ \frac{\sigma_x^2\Delta t^2}{2} & 0 & 0 & \sigma_x^2\Delta t & 0 & 0 \\ 0 & \frac{\sigma_y^2\Delta t^2}{2} & 0 & 0 & \sigma_y^2\Delta t & 0 \\ 0 & 0 & \frac{\sigma_z^2\Delta t^2}{2} & 0 & 0 & \sigma_z^2\Delta t \end{bmatrix} \tag{28b}$$

where Q is the covariance matrix.

3.2. Measurement Model

The non-linear projection equations (Equation (5)) for N cameras satisfy the relation

$$\vec{z} = \vec{h}(\vec{x}) + \vec{w} \tag{29}$$

where

$$\vec{z} = \begin{bmatrix} z_1^x & z_1^z & \dots & z_N^x & z_N^z \end{bmatrix}^T \tag{30}$$

$$\vec{h}(\vec{x}) = \frac{1}{\rho_y} \begin{bmatrix} f_1^x(\rho_x - b_1^x) \\ f_1^z(\rho_z - b_1^z) \\ \vdots \\ f_N^x(\rho_x - b_N^x) \\ f_N^z(\rho_z - b_N^z) \end{bmatrix} \tag{31}$$

$\vec{z}$ denotes the coordinates in pixels of the Center of Projection (COP) of the target relative to the center of the images. $f_1^x, f_1^z, ..., f_N^x, f_N^z$ are the focal lengths in pixels for each camera and each direction separately. $\vec{w}$ is a zero-mean Gaussian measurement noise vector and R is its covariance matrix,

$$E(\vec{w}_k) = 0 \quad , \quad E(\vec{w}_k \vec{w}_k^T) = R \tag{32}$$

3.3. CMKF

The CMKF [23–26] is a less common filter than the EKF and the UKF, and it cannot be implemented on all non-linear systems; only the measurement equations are allowed to be non-linear (the process equations are compelled to be linear) and not all non-linearities can be dealt with. The main idea in the CMKF is to rearrange the non-linear measurement equations into linear equations.

The CMKF is a linear filter, and therefore has a stability proof, which leads to robustness to different initial conditions and to large time increments. On the other hand, the noise in the rearranged measurement equations is not necessarily white, and it may be difficult to determine its statistics. An inaccurate assessment of the noise statistics can result in a biased estimation of the state.

The CMKF fits quite well to the case of relative position estimation using stereovision with no a-priori information. The process equation is linear (Equation (26)) and the non-linearity in the measurement equation (Equation (5)) can be dealt with (Equation (6)).

Equation (6) shows that

$$\begin{pmatrix} 1 & -z_1^x/f_1 & 0 \\ 0 & -z_1^z/f_1 & 1 \\ \vdots & & \\ 1 & -z_N^x/f_N & 0 \\ 0 & -z_N^z/f_N & 1 \end{pmatrix} \vec{\rho}_m = \begin{pmatrix} b_x^1 \\ b_z^1 \\ \vdots \\ b_x^N \\ b_z^N \end{pmatrix} \quad , \quad \vec{\rho}_m = \begin{pmatrix} \rho_x \\ \rho_y \\ \rho_z \end{pmatrix}_m \tag{33}$$

This is a linear system, $A\vec{\rho}_m = \vec{b}$. The unknown vector $\vec{\rho}_m$ can be evaluated by solving Equation (33) using different methods, as discussed in Section 5.

Although the vector $\vec{\rho}_m$ does not necessarily contain additive noise, it was approximated in Section 2.3 using a linear model

$$\vec{\rho}_m = f(\vec{\rho}, \vec{w}) \approx \vec{\rho} + \vec{w} = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \end{bmatrix} \vec{x} + \vec{w} \tag{34}$$

Equation (34) is the CMKF measurement equation, where $\vec{w}$ is a zero-mean measurement noise vector and R is its covariance matrix.

$$E(\vec{w}_k) = 0 \quad , \quad E(\vec{w}_k \vec{w}_k^T) = R = \begin{bmatrix} \sigma_{\rho_x}^2 & 0 & 0 \\ 0 & \sigma_{\rho_y}^2 & 0 \\ 0 & 0 & \sigma_{\rho_z}^2 \end{bmatrix} \tag{35}$$

where σ_{ρ_x}, σ_{ρ_y}, σ_{ρ_z} are approximated using Equation (18),

$$\sigma_{\rho_x} = \frac{\rho_y}{bf}\sqrt{(\rho_x - b_1^x)^2 + (\rho_x - b_2^x)^2}\,\Delta z \quad , \quad \sigma_{\rho_y} = \frac{\rho_y^2}{bf}\sqrt{2}\,\Delta z \quad , \quad \sigma_{\rho_z} = \frac{\rho_y}{bf}\left(\frac{b^2}{2} + 2\rho_z^2\right)^{\frac{1}{2}}\Delta z \tag{36}$$

Δz is the standard deviation of the coordinates of the centroid of the projection of the target onto the image plane, assuming that the standard deviation Δz is the same in the x and z directions in both cameras. The rest of the CMKF equations are standard Kalman Filter (KF) equations.

The full details of the CMKF algorithm are described in Algorithm 1, where P, Q, R are the state covariance, process noise covariance and measurement noise covariance respectively, Φ is the discrete state-transition matrix, K is the gain matrix, k is the time step index and $\vec{\rho}_m$ is an estimation of the LOS vector $\vec{\rho}$.

Algorithm 1 CMKF

Initialization:

$$\vec{\hat{x}}_0 = E[\vec{x}_0]$$

$$P_0 = E[(\vec{x}_0 - \vec{\hat{x}}_0)(\vec{x}_0 - \vec{\hat{x}}_0)^T]$$

while Target is within FOV **do**

Time Propagation:

$$\vec{\hat{x}}_{k+1|k} = \Phi\vec{\hat{x}}_{k|k}$$

$$P_{k+1|k} = \Phi P_{k|k}\Phi^T + Q$$

Create Pseudo-Measurements:

$$\vec{z}_k = (\vec{\rho_m})_k$$

$$H = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \end{bmatrix}$$

Measurement Update:

$$K_k = (P_{k+1|k}H^T)^{-1}(HPH^T + R)$$

$$\vec{\hat{x}}_{k+1|k+1} = \vec{\hat{x}}_{k+1|k} + K_k \cdot (\vec{z}_k - H \cdot \vec{\hat{x}}_{k+1|k})$$

$$P_{k+1|k+1} = (I - K_kH)P_{k+1|k}(I - K_kH)^T + K_kRK_k^T$$

end while

3.4. Filters Initialization

By using the methods described in Section 2.4, the obtained measurements are processed in order to initialize the filters through Equation (6), which can be solved using several methods, e.g., LS, TLS and CVE. These methods solve the overdetermined system of linear equations $A\vec{x} = \vec{b}$, where A is

a matrix with more rows than columns. In the LS approach [6], there is an underlying assumption that all the errors are confined to the observation vector $\vec{b}$; its solution is $\vec{x} = (A^T A)^{-1} A^T \vec{b}$. In the TLS approach [29] it is assumed that there are errors both in $\vec{b}$ and A. The CVE approach was presented in Section 2.3. In order to determine the preferred method of solution, the performance of these methods will be examined in Section 5.

3.5. Estimation Scheme

The estimation scheme in Figure 3 describes the implemented algorithm. Every step, a measurement is acquired. If a target has not been found yet, or the algorithm was initialized in the last step, then a target recognition algorithm is activated. Only if a target was found by the target recognition algorithm, the different filters are initialized. After each filtering step, the algorithm checks if the estimation is reliable. Estimation is declared as non-reliable if the produced state is not reasonable in relation to the laboratory dimensions or to the state of the previous step. A non-reliable state can also be produced in case there has been too few matched feature points for several sequential steps. This happens mostly if the target has left the scene. If the state is unreliable, the algorithm is re-initialized, meaning that the filters are initialized and the target has to be found again.

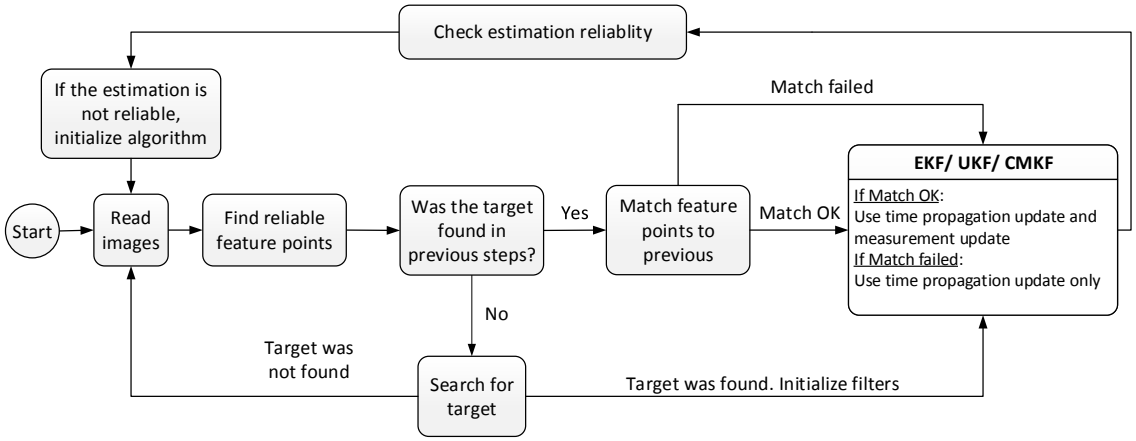


Figure 3. The estimation logic. This scheme consists of a filter re-initialization step, which is activated in case the estimation is no longer reliable.

4. Target Tracking

The LOS between the chaser and the target is the vector $\vec{\rho}$, defined in previous sections. It determines whether the target is visible to the chaser.

The direction of the LOS can be expressed using azimuth and elevation angles. In general, the tracking algorithm has to keep these two angles bounded. In the two dimensional case, only the azimuth angle has to be controlled. The bounds of these angles are determined by the camera rig's FOV, which is determined by the cameras' properties and the geometric arrangement.

4.1. FOV Azimuth Angle Bounds

Figure 4 depicts two vertically aligned and horizontally positioned cameras, and their combined FOV, denoted by *StereoView*. α is the horizontal Angle-of-View (AOV), *baseline* is the distance between the cameras and *Range* is the distance between the cameras and the target. The following relation can be inferred:

$$StereoView = 2 \tan\left(\frac{\alpha}{2}\right) Range - baseline \tag{37}$$

It can be seen that the use of stereovision diminishes the combined FOV. More specifically, enlargement of the baseline leads to a smaller *StereoView*. On the other hand, increasing the baseline

also leads to a higher accuracy (Equation (18)). By substituting $StereoView = 0$ into Equation (37), The minimal $Range$ is calculated.

$$Range_{min} = \frac{baseline}{2\tan\left(\frac{\alpha}{2}\right)} \tag{38}$$

The horizontal AOV of each camera is expressed as

$$\alpha = 2\arctan\left(\frac{SensorWidth}{2f}\right) = 2\arctan\left(\frac{SensorWidth}{2R_F f_{base}}\right) \tag{39}$$

where f is the focal length. f_{base} and R_F are the basic focal length and resize factor, respectively, as defined in Section 2.2. $SensorWidth$ is the width of the optical sensor.

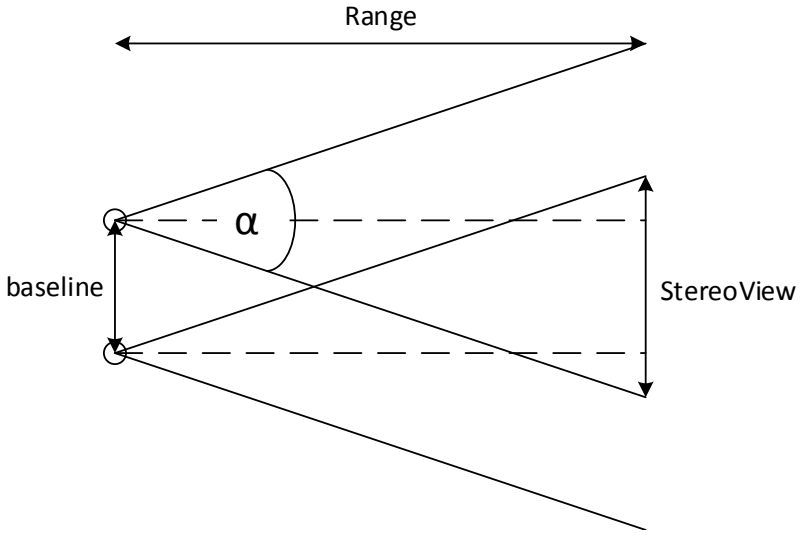


Figure 4. A two dimensional geometry of two aligned cameras with their combined FOV.

Figure 5 depicts $StereoView$ as defined in Equation (37). ϕ is the azimuth angle of the LOS vector $\vec{\rho}$. In order to maintain the target within $StereoView$, ϕ_{max} is defined as the maximal allowed azimuth angle for a certain $Range$. From the geometry depicted in Figure 5, ϕ_{max} is calculated,

$$\begin{gathered} |\phi| \leq \phi_{max} \\ \phi_{max}(Range) = \arctan\left(\frac{StereoView/2}{Range}\right) \end{gathered} \tag{40}$$

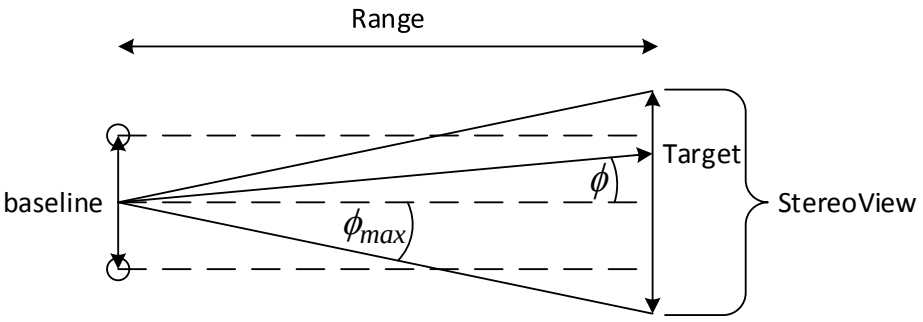


Figure 5. A two dimensional geometry of two aligned cameras with their horizontal angle of view bound.

4.2. Tracking Algorithm

Figure 6 depicts the chaser, the target and the LOS vector $\vec{\rho}$. $\vec{x}_c^{\mathcal{I}}$ and $\vec{x}_t^{\mathcal{I}}$ are the chaser and target's inertial position vectors, respectively. θ_c is the chaser's body angle. $\mathcal{C}$ is a cartesian right-hand body-fixed reference frame attached to the chaser. $\vec{x}^{\mathcal{C}}$, $\vec{y}^{\mathcal{C}}$, $\vec{x}^{\mathcal{I}}$ and $\vec{y}^{\mathcal{I}}$ are the principal axes of the chaser's body frame and the inertial frame, respectively. Assume that the camera rig is installed on the chaser along the $\vec{y}_{\mathcal{C}}$ direction. ϕ is the azimuth angle and is calculated as

$$\phi = \tan^{-1}\left[\frac{(\vec{x}^{\mathcal{C}})^T\vec{\rho}^{\mathcal{C}}}{(\vec{y}^{\mathcal{C}})^T\vec{\rho}^{\mathcal{C}}}\right] = \tan^{-1}\left(\frac{\rho_x^{\mathcal{C}}}{\rho_y^{\mathcal{C}}}\right) \tag{41}$$

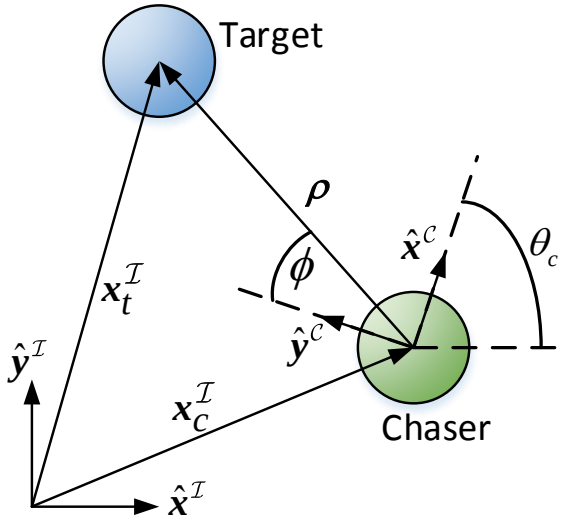


Figure 6. Top view of a chaser and a target, showing the azimuth angle.

Figure 7 depicts the LOS control algorithm. In the tracking algorithm, it is desired to maintain the target in the FOV center, or in other words, to minimize $|\phi|$. It is done by a Proportional Derivative (PD) controller. The signals $\vec{x}_c^{\mathcal{I}}$ and $\vec{x}_t^{\mathcal{I}}$ are constantly changing and, therefore, ϕ will not converge to zero.

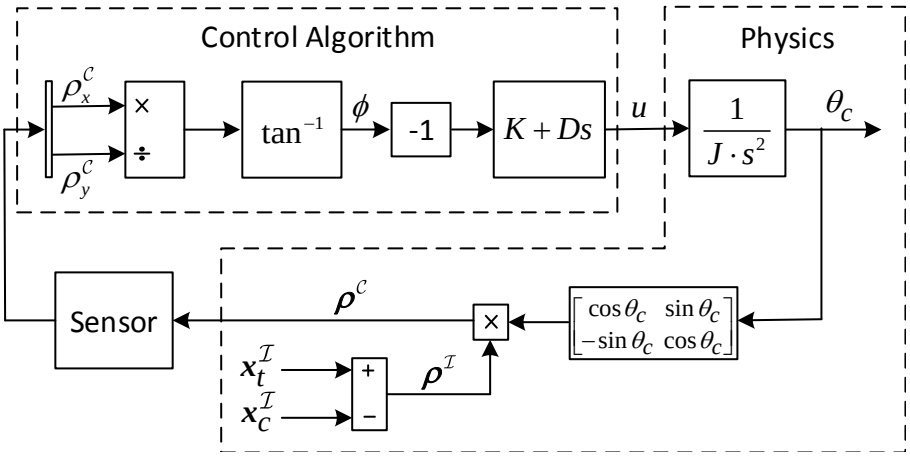


Figure 7. The LOS Control Algorithm.

In Section 3.5 it was mentioned that there are cases where the state is unreliable and the estimation algorithm is initialized. In that case, there is no estimation of the vector $\vec{\rho}$, and the tracking algorithm

cannot be implemented. For the sake of simplicity, in those cases, a default reference value of 0 degrees is used for θ_c, which makes the robot look straight ahead to the direction of the target, assuming it is not moving.

4.3. Tracking Algorithm Stability

The closed-loop dynamics in Figure 7 can be written as

$$\ddot{\theta}_c + \frac{D}{J}\frac{(\rho_x^{\mathcal{I}})^2 + (\rho_y^{\mathcal{I}})^2}{(\rho_y^{\mathcal{I}}\cos(\theta_c) - \rho_x^{\mathcal{I}}\sin(\theta_c))^2}\dot{\theta}_c + \frac{K}{J}\frac{\rho_x^{\mathcal{I}}\cos(\theta_c) + \rho_y^{\mathcal{I}}\sin(\theta_c)}{\rho_y^{\mathcal{I}}\cos(\theta_c) - \rho_x^{\mathcal{I}}\sin(\theta_c)} = 0 \tag{42}$$

where K and D are the PD controller proportional and derivative gains, respectively. J is the polar moment of inertia of the chaser. Notice that the properties of the chaser are assumed to be known. The stability analysis will be carried out around an equilibrium where the chaser is looking directly at the target. For simplicity, it is also assumed that the chaser and the target share the same $x^{\mathcal{I}}$ coordinate. In other words,

$$(\rho_x^{\mathcal{I}})_{eq} = 0 \quad ; \quad (\theta_c)_{eq} = 0 \tag{43}$$

At this equilibrium, Equation (42) yields

$$\ddot{\theta}_c + \frac{D}{J}\dot{\theta}_c + \frac{K}{J}\theta_c = 0 \tag{44}$$

It can be seen that for $K, D > 0$ ($J > 0$ because it is a moment of inertia), Equation (44) represents an asymptotically stable system.

This system is also asymptotically stable for a more general case, where the chaser is looking directly to the target, but the LOS vector is not aligned with the inertial coordinate system $\mathcal{I}$, that is,

$$(\rho_x^{\mathcal{I}})_{eq} \neq 0 \quad ; \quad (\theta_c)_{eq} \neq 0 \quad ; \quad (\rho_x^{\mathcal{C}})_{eq} = 0 \quad ; \quad \phi_{eq} = 0 \tag{45}$$

In this case, in order to prove stability, it is required to define a new inertial coordinate frame, which is rotated by $(\theta_c)_{eq}$ with respect to $\mathcal{I}$. In the rotated inertial coordinate frame, Equation (43) is satisfied and the stability proof holds.

5. Numerical Study

To initialize the filters (Section 3.4) and the distance value for each feature point (Section 2.4.3), it is required to solve Equation (6). In this section, a numerical evaluation will compare the solutions of Equation (6) using the LS, TLS and CVE approaches.

5.1. Static Problem Description

Consider a static object, which is viewed by a camera rig with either 2 or 4 cameras. In the case of 2 cameras, the cameras are vertically aligned and horizontally positioned with a baseline of length b. In the case of 4 cameras, the cameras are located at the vertices of a square with edges of length b.

Each camera acquires measurements of the viewed object according to Equation (5). Each measurement is corrupted with a zero mean Gaussian noise with a standard deviation ΔZ and is rounded to the closest integer (because the measurements are in pixels). The noisy measurements are then used for writing Equation (6), which is solved using LS and TLS. In the case of 2 cameras, CVE is also used.

This routine is carried out in 5000 Monte-Carlo runs, which produce estimations for the relative position of the object relative to the camera rig. RANSAC [20] was used in order to detect and reject outliers. Consequently, approximately 4% of the estimations were considered outliers and were discarded. The parameter values used in this numerical estimation are summarized in Table 1.

Table 1. Simulation parameter values.

Parameter	Value	Units
$f = f_{eff}$	750	px
ρ_x	1	m
ρ_y	5	m
ρ_z	1	m
$b = baseline$	0.16	m
ΔZ	5	px

5.2. Results

Tables 2 and 3 provide the statistical properties of the histograms depicted in Figures 8 and 9, respectively. μ and σ denote the mean value and standard deviation, respectively. $(\sigma_{LS})_{Est}$ is the estimation of the standard deviations of ρ_x, ρ_y, ρ_z in the case of 2 cameras, which is estimated using Equation (18).

Although the estimation bias is not negligible with LS, TLS and CVE with 2 or 4 cameras, TLS and CVE produce significantly less bias than LS. Also, the use of 4 cameras does not always yield less bias compared to 2 cameras.

Table 2. Monte-Carlo results summary, 2 cameras.

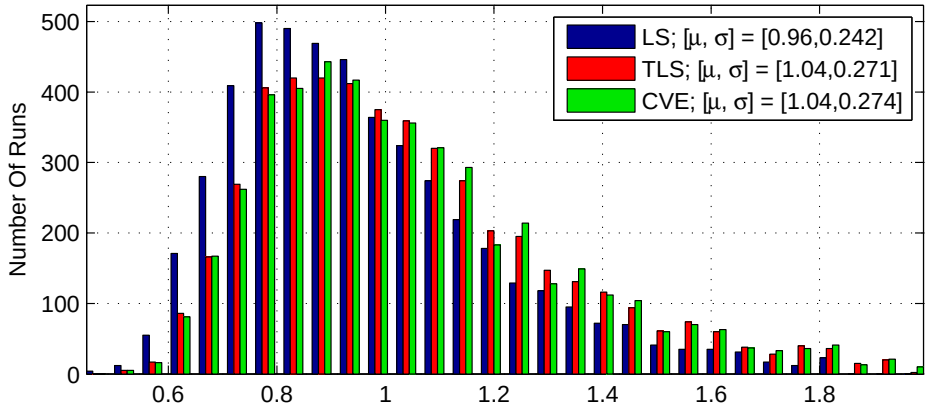
Parameter	True Value	μ_{LS}	μ_{TLS}	μ_{CVE}	σ_{LS}	$(\sigma_{LS})_{Est}$	σ_{TLS}	σ_{CVE}
ρ_x [m]	1	0.96	1.04	1.04	0.242	0.272	0.271	0.274
ρ_y [m]	5	4.78	5.27	5.29	1.31	1.473	1.58	1.58
ρ_z [m]	1	0.947	1.04	1.04	0.249	0.295	0.297	0.29

Table 3. Monte-Carlo results summary, 4 cameras.

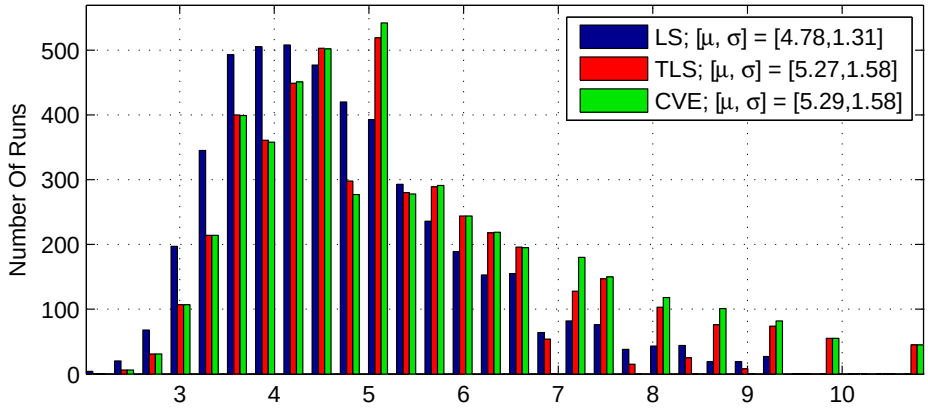
Parameter	True Value	μ_{LS}	μ_{TLS}	σ_{LS}	σ_{TLS}
ρ_x [m]	1	0.917	1.01	0.106	0.129
ρ_y [m]	5	4.54	5.03	0.565	0.702
ρ_z [m]	1	0.901	1.01	0.123	0.152

In the y direction, TLS and CVE produce slightly greater dispersion than LS. As expected, the use of 4 cameras yields less dispersion than 2 cameras. $(\sigma_{LS})_{Estimation}$ is a fair approximation of σ_{LS} because their corresponding components share the same magnitude.

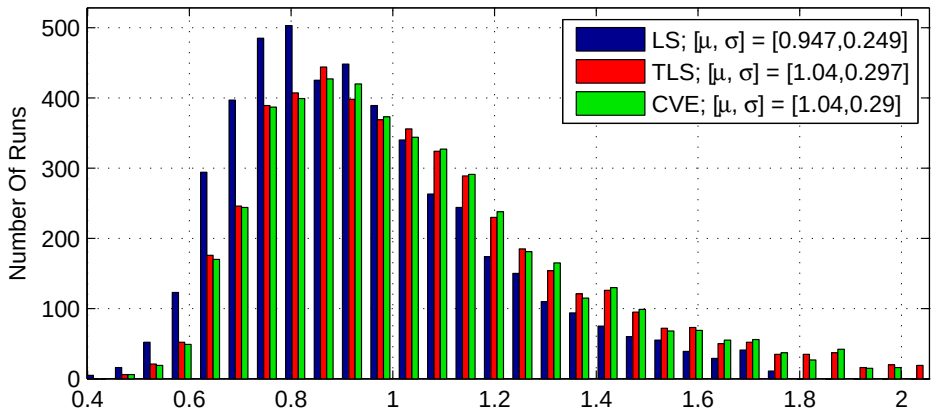
According to these results, for the case of 2 cameras, LS, TLS, and CVE have similar performance. In this research TLS is used. For the case of 4 cameras, TLS is preferable over LS because of the bias improvement, while the dispersion is only slightly greater.



(a) Estimation with 2 cameras of ρ_x [m]



(b) Estimation with 2 cameras of ρ_y [m]



(c) Estimation with 2 cameras of ρ_z [m]

Figure 8. Monte carlo results of a static LOS estimation using stereovision measurements with 2 cameras.

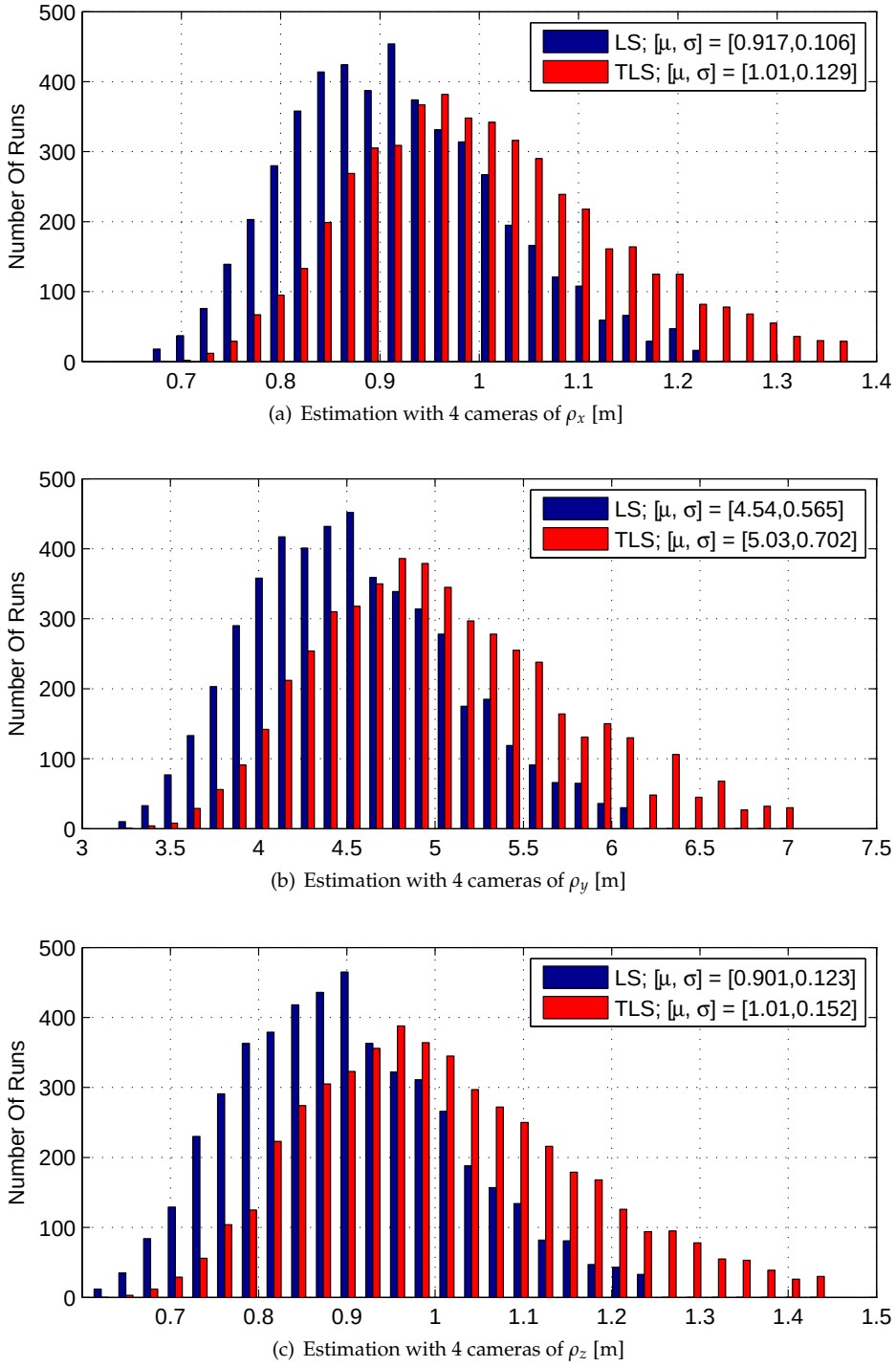


Figure 9. Monte carlo results of a static LOS estimation using stereovision measurements with 4 cameras.

6. Experimental Results

The algorithms developed herein were tested in a series of experiments. These algorithms include image acquisition software, and control software, which achieve real-time performance. As mentioned in Section 1, the experimental part includes 2 cameras.

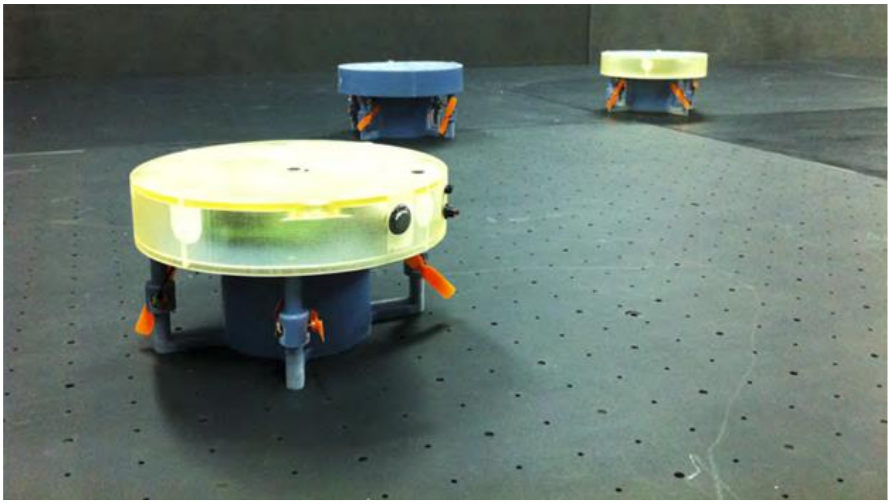
The sampling frequency of the computer-vision software is mainly dependent on the number of discovered feature points, and the size of the target projection in the frames. As a result,

the computer-vision software's time step varies between 0.4 and 0.6 s. The control software operates at a constant sampling frequency of 30 Hz.

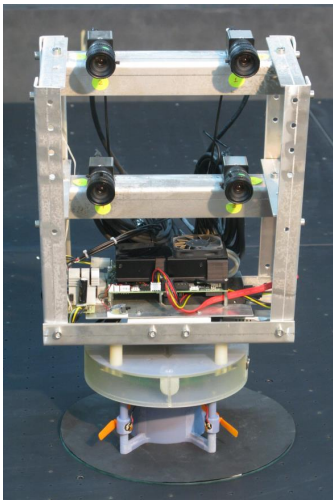
The initial state covariance for all the filters is a 6×6 diagonal matrix, whose main diagonal is $[4\ \mathrm{m}^2, 4\ \mathrm{m}^2, 4\ \mathrm{m}^2, 225\ \mathrm{m}^2/\mathrm{s}^2, 225\ \mathrm{m}^2/\mathrm{s}^2, 225\ \mathrm{m}^2/\mathrm{s}^2]$. The values for $[\sigma_x, \sigma_y, \sigma_z]$ for the process noise covariance, as mentioned in Section 3.1, are [1,1,0] m/s^2, [0.5,0.1,0] m/s^2 and [0.8,0.8,0] m/s^2 for the EKF, UKF, and for the CMKF, respectively. Because only 2 cameras are used, the measurement noise covariance for the EKF and UKF is a 4×4 diagonal matrix. Its main diagonal is $[25\ \mathrm{px}^2, 100\ \mathrm{px}^2, 25\ \mathrm{px}^2, 100\ \mathrm{px}^2]$ For the CMKF, the measurement noise covariance is calculated in each step, as described in Section 3.3. For that, ΔZ is assumed to be 10 pixels. Also, the UKF algorithm in this research uses a simple uniform set of weights [13].

6.1. Laboratory Description

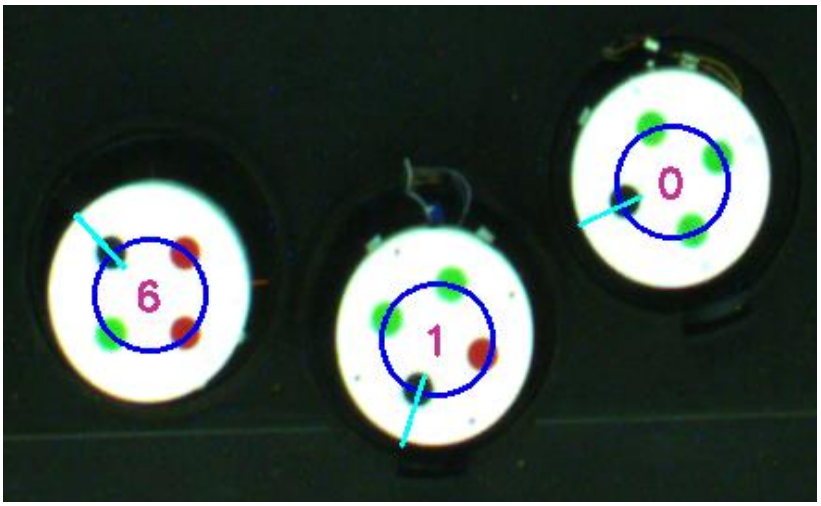
The experimental part of this research was conducted in the DSSL at the Technion. The DSSL is equipped with an air table, floating robots (Figure 10a), communication system, main computer, a stereovision camera rig (Figure 10b) and an overhead camera (Figure 10c), which measures the position and orientation of the robots on the air table.



(a) Three robots placed on the air table.



(b) Chaser equipped with a stereovision system.



(c) Image taken by the overhead camera's software. Each robot is recognized in the scene, including its orientation and its velocity direction.

Figure 10. The equipment in the DSSL.

Each robot is characterized by four circles with different colors in different order (Figure 10c). The overhead camera uses those colors to identify each robot and determine its orientation. The overhead camera is calibrated so as to relate the position of each robot in the image plane to its corresponding inertial position.

Figure 11 depicts the chaser with the camera rig facing the target. As can be seen in Figure 11, the camera rig contains four cameras, which are all aligned. Notice that although the mounted camera rig does include four camera, the experimental part of the research utilizes only two of them (the top two, which are marked as "Cam1" and "Cam2" in Figure 11).

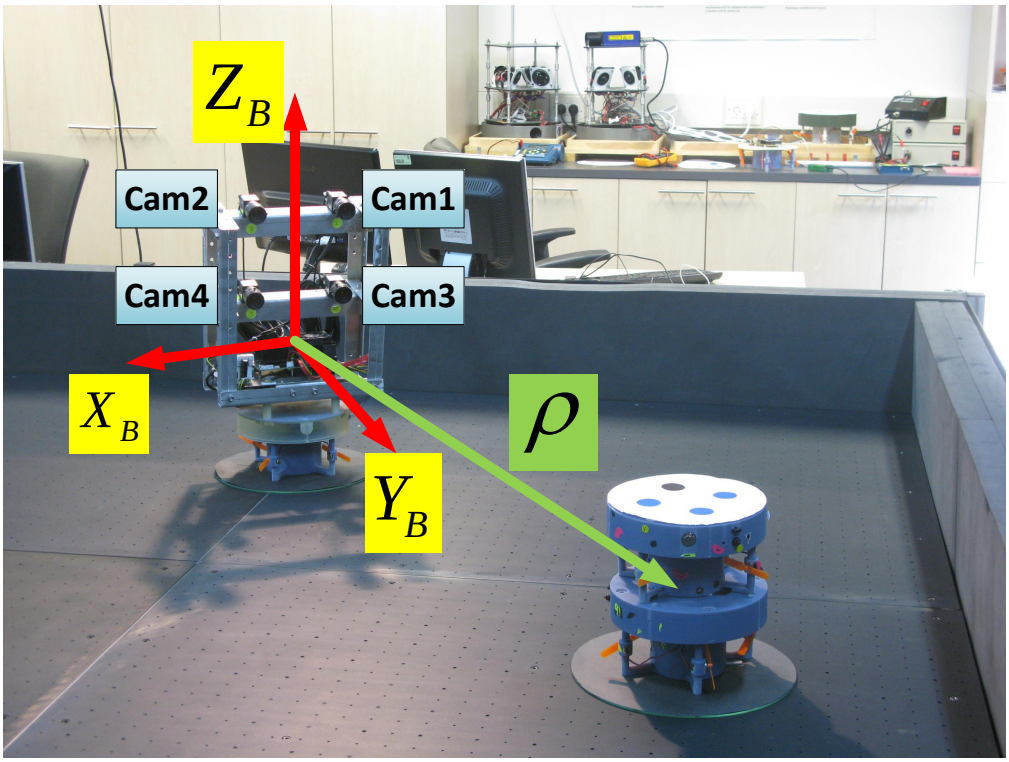


Figure 11. The body frame, camera numbers and LOS vector, shown in the laboratory environment.

6.2. Reference Signals

In the experiments, the values $\rho_x^{\mathcal{C}}$, $\rho_y^{\mathcal{C}}$, ϕ, θ_c are estimated and compared to reference values, where the corresponding reference values are calculated in the following manner. The overhead camera (see Section 6.1) acquires measurements of the position of the chaser and the target in the inertial frame $\vec{x}_c^{\mathcal{I}}$, $\vec{x}_t^{\mathcal{I}}$, and the orientation of the chaser θ_c. $\rho_x^{\mathcal{C}}$ and $\rho_y^{\mathcal{C}}$ are calculated using the equation

$$\vec{\rho}^{\mathcal{C}} = \begin{bmatrix} \rho_x^{\mathcal{C}} \\ \rho_y^{\mathcal{C}} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \vec{\rho}^{\mathcal{I}} \tag{46}$$

The LOS azimuth angle ϕ is calculated using Equation (41).

6.3. FOV Properties

The parameter values used in this research are given in Table 4. Using Equations (37), (38), (40) and Table 4, *StereoView* and $Range_{min}$ are calculated,

$$StereoView = 0.73Range - 0.16\ [\mathrm{m}] \tag{47}$$

$$Range_{min} = 0.218\ \mathrm{m} \tag{48}$$

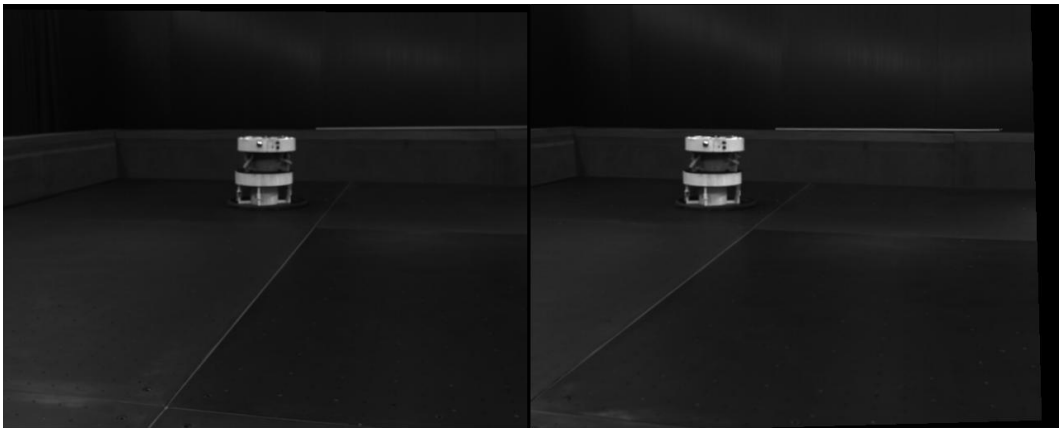
$$\phi_{max}(Range) = \arctan\left(0.367 - \frac{0.08\,[\mathrm{m}]}{Range}\right) \tag{49}$$

Table 4. FOV Properties.

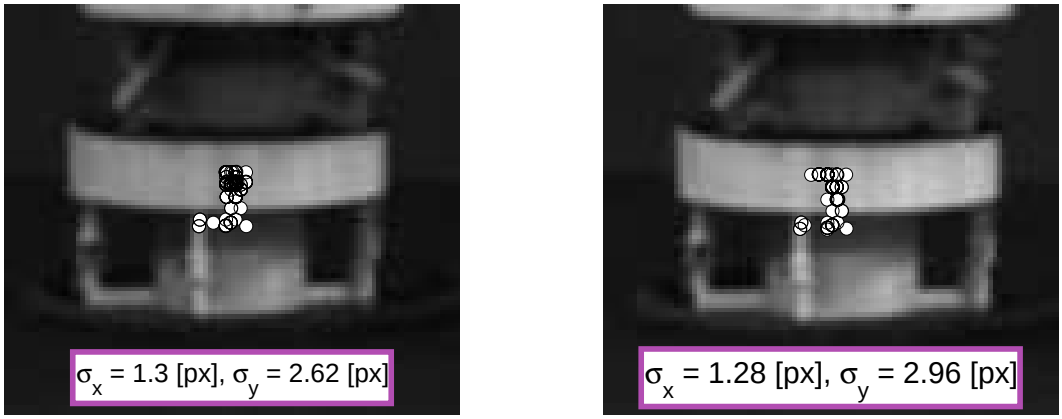
Parameter	Value	Units
$SensorWidth$	6.784	mm
f_{base}	8	mm
R_F	0.5	
α	40.3	deg
$baseline$	0.16	m

6.4. Static Experiment

A static experiment was conducted, wherein the target and chaser are placed at fixed locations. The purpose of this experiment is to test the target recognition algorithm. Due to measurement noise, which is manifested by a different number of feature points and locations, the target's COP in each frame is slightly different, as seen in Figure 12. It is desired to measure this dispersion in order to use it in the estimation algorithm.



(a) The images taken by the two cameras



(b) Magnification of the images taken by the two cameras, with marks of the COP in all the frames.

Figure 12. The images taken by the two cameras, with a statistical analysis of the COP dispersion.

In this experiment, due to the prominence of the color of the target's bottom compared to its top, a larger number of features was found at the bottom. Consequently, the COP location was estimated to

be lower than expected. As seen in Equation (15), this has minor to no effect on the components of the LOS estimation in the x and y directions. The main influence on these components is the COP's horizontal position. As noted in Figure 13, the dispersion of the target's COP is

$$\sigma_x \approx 1.3\,\text{px} \qquad \sigma_y \approx 2.8\,\text{px} \tag{50}$$

Although the dispersions in both directions share the same magnitude, it is reasonable to assume that σ_y is greater than σ_x due to the target's shape, which is characterized by a slightly longer vertical dimension.

Figures 13 and 14 depict the LOS horizontal components and the azimuth estimation, respectively. It can be seen that the estimation of ρ_x and ρ_y, and consequently ϕ, are biased, and that all of the estimated values are constantly over-estimated; that is, the estimated values are larger than the true values. The bias values are approximately

$$bias(\rho_x) \approx 0.5\,\text{cm} \qquad bias(\rho_y) \approx 30\,\text{cm} \qquad bias(\phi) \approx 0.25\,\text{deg} \tag{51}$$

It can also be seen that in the static estimation scenario, due to the slight changes in the target's COP in each time step, the estimates of the filters do not converge, but stay around the average value. By using these results, it is not definite which estimator works best. The use of the CMKF did not produce better results than the EKF and UKF.

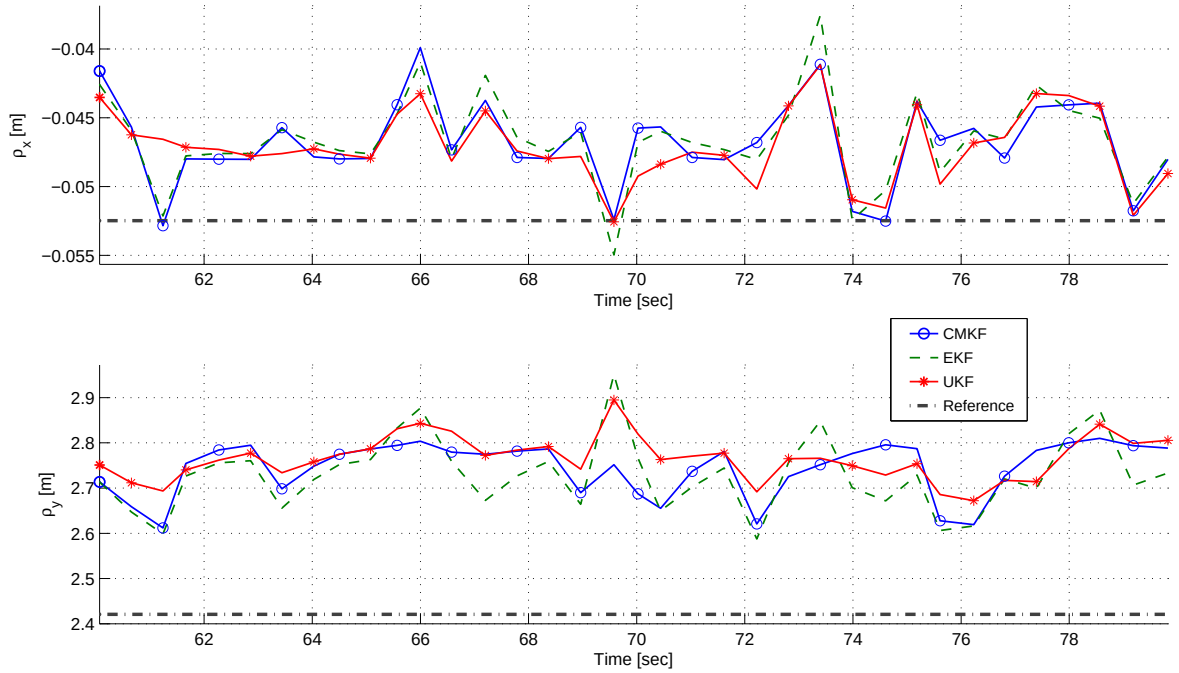


Figure 13. A comparison of ρ_x (**top**) and ρ_y (**bottom**) estimation using different filters in the static experiment.

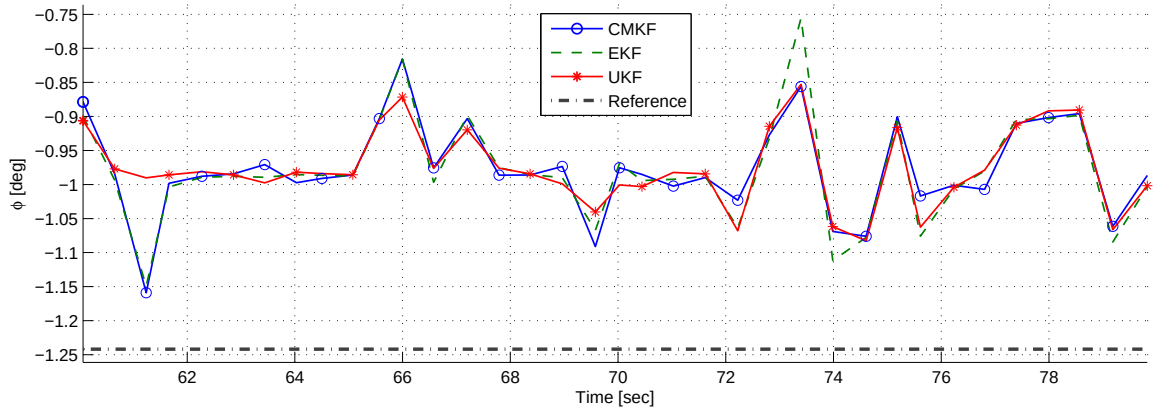


Figure 14. A comparison of the azimuth estimation using different filters in the static experiment.

6.5. Semi-Static Experiment

A semi-static experiment was conducted where the target is stationary and the chaser had to maintain its position and orientation in order to keep the target within its FOV, while dealing with dynamic modeling inaccuracies and external disturbances.

Three impulses of external torques were applied during this experiment. These torques are marked in the different figures.

Figure 15 depicts the chaser's body angle θ_c. It can be seen that after each impulse of external torque, θ_c converges to its nominal value with a characteristic response of a second-order system with two complex stable poles. This is expected, based on the rotational closed-loop dynamics modeled in Section 4.2.

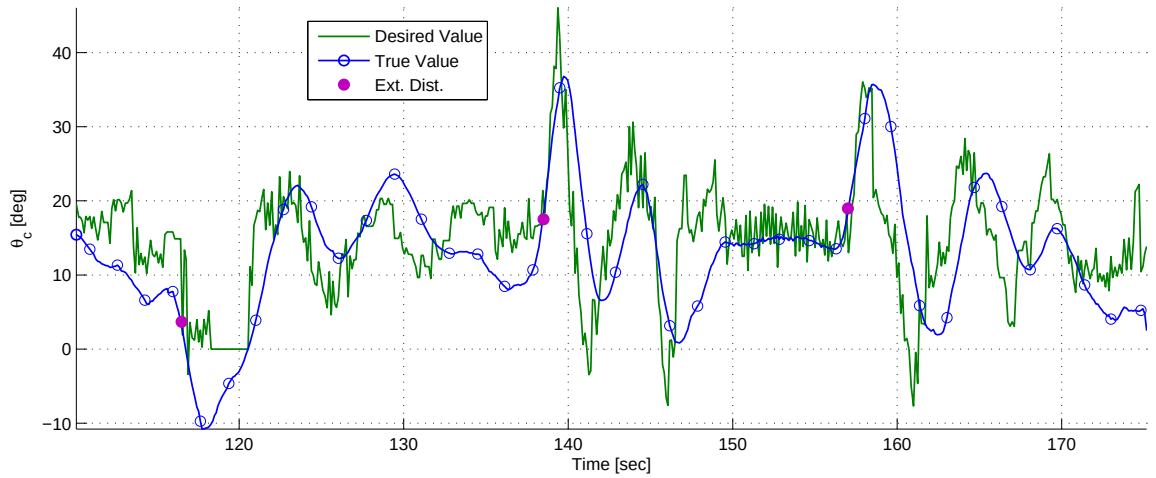


Figure 15. A comparison between the true body angle and the desired body angle.

Between 116 and 120 s, the target left the chaser's FOV. As a result, θ_c's desired value is zero. While the chaser rotated towards $\theta_c = 0$, the target returned to the FOV, the target recognition algorithm detected it and tracking was resumed.

Figure 16 depicts the LOS components estimation as well as the reference values, where the reference values calculation is described in Section 6.2. The most prominent property in these graphs is the bias in the ρ_y estimation. Most of the time this bias has a constant mean value with oscillations and occasional peaks. There is also bias in the ρ_x estimation, but unlike the ρ_y bias, the ρ_x bias has negligible oscillations. Also, a good correspondence between the ρ_x estimation and its reference is notable.

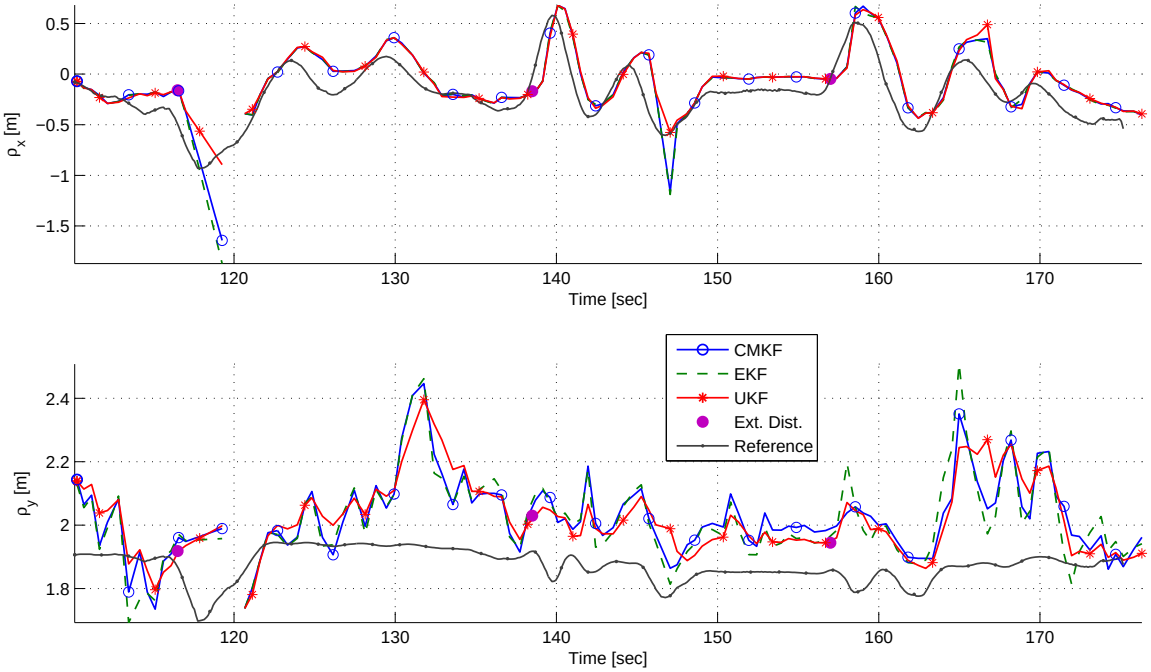


Figure 16. A comparison of ρ_x (**top**) and ρ_y (**bottom**) estimation using different filters in the semi-static experiment.

The absence of the target in the chaser's FOV is seen in approximately 116–120 s and shortly after this event, the estimators diverge. In these sorts of events, the control algorithm is programmed to ignore the unreliable current estimate, and calculate the control signals according to a nominal predefined state. This divergence event ended when the target entered the FOV and tracking was resumed.

Figure 17 depicts the azimuth angle ϕ and its reference. ϕ is calculated using ρ_x and ρ_y, and, therefore, all the different phenomena that occurred in ρ_x and ρ_y are reappearing in ϕ as well. It is also notable that the ϕ estimation is characterized by bias and delay.

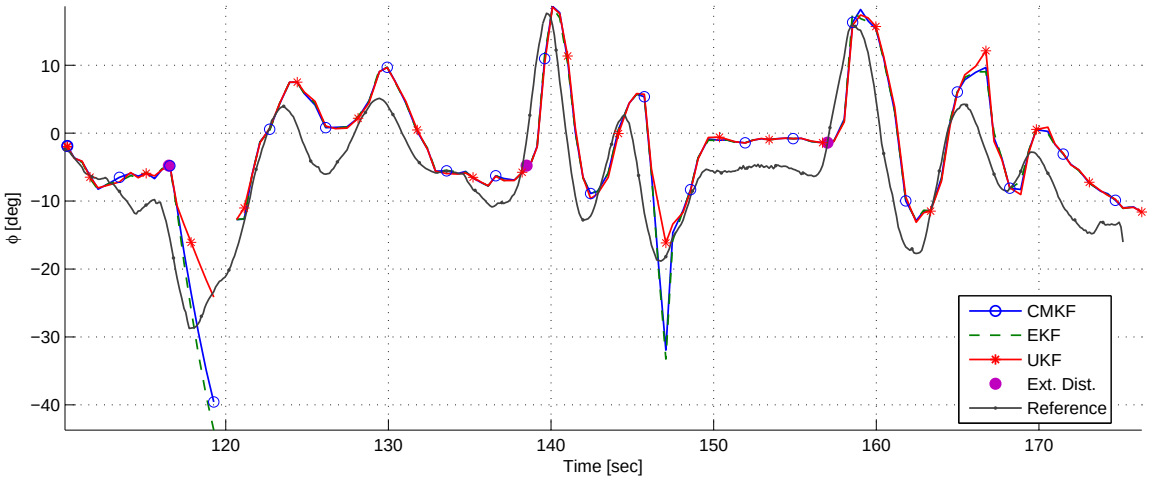


Figure 17. A comparison of the azimuth estimates using different filters in the semi-static experiment.

At $t = 146$, it can be seen that the CMKF and the EKF estimates had a large error compared to the UKF estimate. It is important to note that in this experiment the control signals were calculated using the UKF output.

Figure 18 depicts the torque commands versus the applied torque. As seen, the thrusters provide a discrete control torque although the commanded value is continuous.

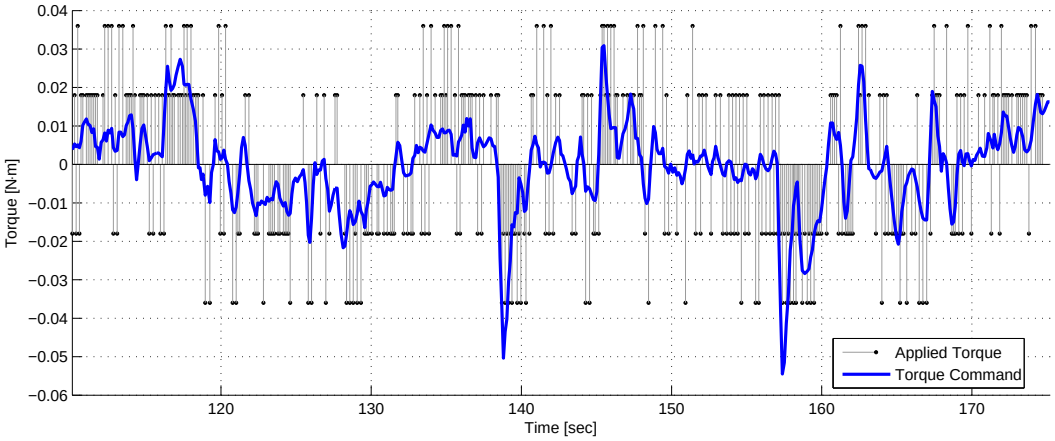


Figure 18. A comparison between the continuous torque command and the discrete applied torque.

6.6. Dynamical Experiment

A dynamical experiment was conducted, wherein the target moved in a cyclic trajectory. The chaser's objective was to maintain its position and change its orientation in order to keep the target within its FOV.

Figure 19 depicts the trajectories of the chaser and the target during the experiment. In order to better illustrate the experiment, four points in time were selected. For each, the position and orientation of the chaser and the position of the target were marked.

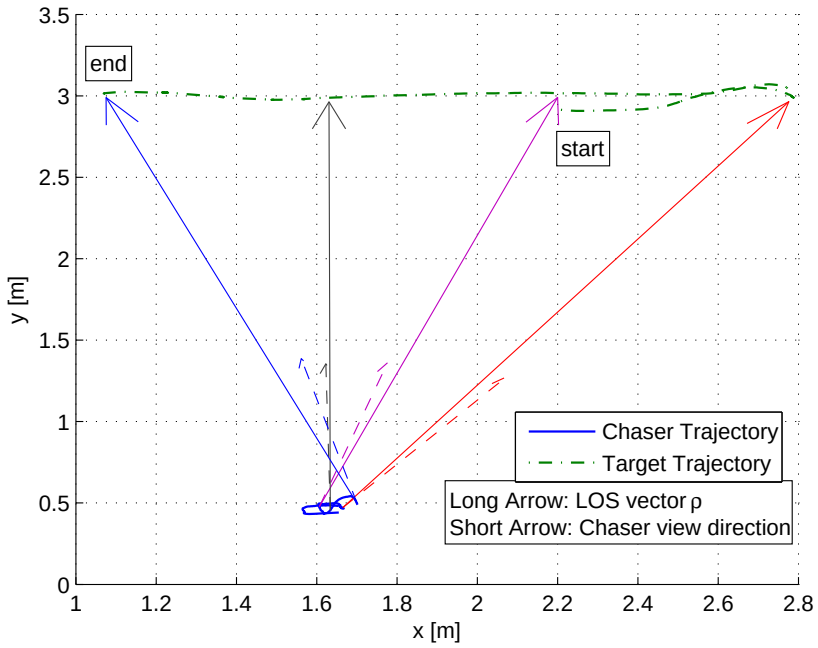


Figure 19. The trajectory of the target and the chaser including the LOS and body directions at different time steps.

The long arrows are the LOS vectors in different time steps, and the short arrows are the directions of the chaser's body in each time step.

Figure 20 depicts the LOS components estimation and references. A bias is seen in the ρ_y estimation. It is clearly seen that the EKF has the worst performance. On the other hand, the ρ_x estimation has good correspondence to its reference with all estimators.

Figure 21 depicts the azimuth angle ϕ during the experiment. Although ρ_y was constantly over-estimated, it can be seen that the azimuth angle has a good correspondence to its reference without bias.

During approximately 190–194 s and 227–231 s, a large error is seen in the azimuth angle. These large errors had likely occurred due to rapid and substantial changes in the relative position. It is important to remember that the estimation algorithm has an approximately half a second delay.

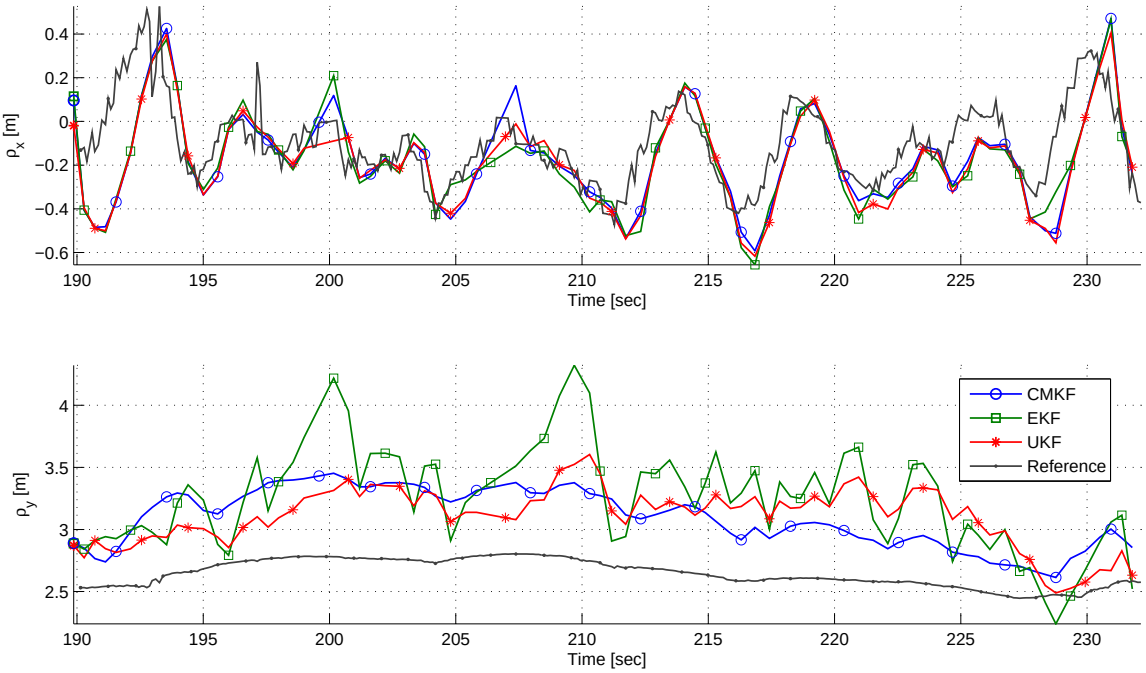


Figure 20. A comparison of ρ_x (**top**) and ρ_y (**bottom**) estimation using different filters in the dynamic experiment.

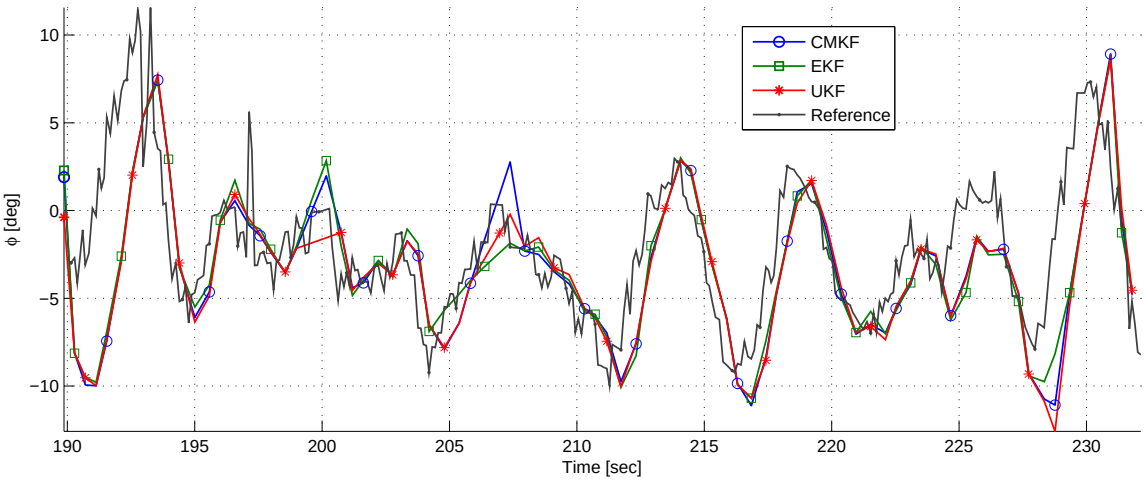


Figure 21. A comparison of the azimuth estimates using different filters in the dynamic experiment.

Figure 22 depicts the torque commands versus the applied torque. As seen, the control signal oscillates, which implies that the dynamical system does not reach an equilibrium. This happens because as long as the target continues moving, the chaser has to keep adjusting its orientation.

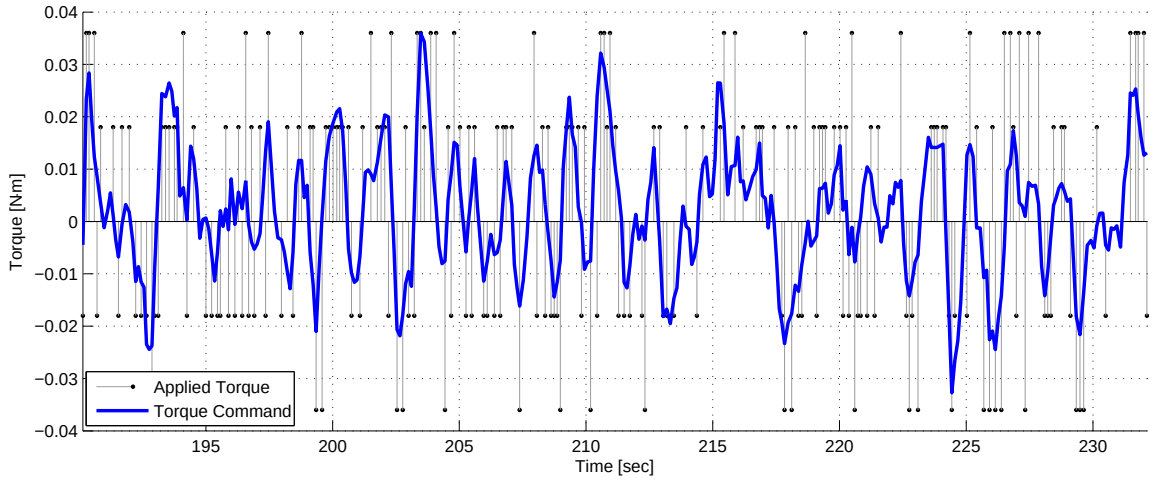


Figure 22. A comparison between the continuous control torque signal, the discrete applied torque and the PD signal.

6.7. Error Analysis

The results in Sections 6.4–6.6 exhibit a bias in the LOS depth component's estimation. It seems that for stereovision with two cameras, bias is an inherent property of the system. This bias was also seen in Figure 8 and in previous work [10].

An interesting issue worth discussing is the fact that although the LOS depth component ρ_y is biased, the estimated azimuth angle ϕ is quite accurate. Recall that

$$\phi = \tan^{-1}\left(\frac{\rho_x}{\rho_y}\right) \approx \frac{\rho_x}{\rho_y} \tag{52}$$

and hence

$$\Delta\phi \approx \sqrt{\left(\frac{\partial\phi}{\partial\rho_x}\Delta\rho_x\right)^2 + \left(\frac{\partial\phi}{\partial\rho_y}\Delta\rho_y\right)^2} \approx \frac{1}{\rho_y}\Delta\rho_x + \frac{|\rho_x|}{\rho_y^2}\Delta\rho_y \qquad (\rho_y > 0) \tag{53}$$

Equation (53) shows that the errors $\Delta\rho_x$, $\Delta\rho_y$ contribute to the error $\Delta\phi$, but they are mitigated by ρ_y and ρ_y^2, respectively. Since ρ_y has a relatively large value, it reduces the error effect dramatically.

7. Conclusions

In this research, a stereovision based algorithm for detection of non-cooperative targets and estimation of the relative position in real-time was developed. The detection algorithm does not require any a priori information except the assumption that the target is the most dominant object in the image. Moreover, a real-time tracking algorithm, which utilizes the stereovision information and keeps the target within the FOV, was developed.

A numerical study, which studies solution methods for the measurement equations was performed, and experiments which utilized the different algorithms were conducted. In these experiments, the performance of three estimators was compared using experimental data.

Real-time performance of the estimation and LOS control algorithms was demonstrated in the experiments carried at in the DSSL. The experimental and numerical results show that there is a non-negligible bias in the LOS depth component estimation with all of the estimators. On the other hand, the LOS horizontal component is estimated with a much smaller bias. The semi-static experiment

exhibited the target detection algorithm performance in case where the target leaves and returns to the FOV.

Although the three estimators have similar behaviour, it seems that the EKF has the poorest performance. Deciding which estimator is better, the UKF or the CMKF, is not trivial, because most of the time their performance is similar. On the other hand, in the semi-static experiment, it is shown that unlike the CMKF, the UKF manages to cope with rapid and substantial changes in the dynamics, and therefore it outperforms the CMKF.

By using SURF feature points, simplifying the detection algorithm and using a small state vector, the average time step duration is reduced compared to previous work to approximately 0.4 s, which for most applications, is approximately the upper bound for real time operation.

Author Contributions: Tomer Shtark performed the analysis and experiments, while Pini Gurfil conceived the research problem and guided the research, performed as part of Tomer Shtark's MSc studies at Technion.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fasano, G.; Grassi, M.; Accardo, D. A stereo-vision based system for autonomous navigation of an in-orbit servicing platform. In Proceedings of the AIAA Infotech@Aerospace Conference, Seattle, WA, USA, 6–9 April 2009.
2. Woffinden, D.C.; Geller, D.K. Relative angles-only navigation and pose estimation for autonomous orbital rendezvous. *J. Guid. Control Dyn.* **2007**, *30*, 1455–1469.
3. Terui, F.; Kamimura, H.; Nishida, S. Motion estimation to a failed satellite on orbit using stereo vision and 3D model matching. In Proceedings of the 2006 9th International Conference on Control, Automation, Robotics and Vision, ICARCV'06, Singapore, 5–8 December 2006; pp. 1–8.
4. Segal, S.; Carmi, A.; Gurfil, P. Vision-based relative state estimation of non-cooperative spacecraft under modeling uncertainty. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2011; pp. 1–8.
5. Segal, S.; Carmi, A.; Gurfil, P. Stereovision-based estimation of relative dynamics between noncooperative satellites: Theory and experiments. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 568–584.
6. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*; John Wiley & Sons: New York, NY, USA, 2004.
7. Lichter, M.D.; Dubowsky, S. Estimation of state, shape, and inertial parameters of space objects from sequences of range images. In Proceedings of the Photonics Technologies for Robotics, Automation, and Manufacturing, Providence, RI, USA, 27 October 2003; pp. 194–205.
8. Lichter, M.D.; Dubowsky, S. State, shape, and parameter estimation of space objects from range images. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004), New Orleans, LA, USA, 26 April–1 May 2004; Volume 3, pp. 2974–2979.
9. Sogo, T.; Ishiguro, H.; Trivedi, M.M. Real-time target localization and tracking by n-ocular stereo. In Proceedings of the IEEE Workshop on Omnidirectional Vision, Hilton Head, SC, USA, 12 June 2000; pp. 153–160.
10. Jigalin, A.; Gurfil, P. Vision-Based Relative State Estimation of Unknown Dynamic Target Using Multiple Stereo Rigs. Master's Thesis, Technion—Israel Institute of Technology, Haifa, Israel, 2013.
11. Wan, E.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000 (AS-SPCC), Lake Louise, AB, Canada, 1–4 October 2000; pp. 153–158.
12. Julier, S.J.; Uhlmann, J.K. New extension of the Kalman filter to nonlinear systems. In Proceedings of the AeroSense'97. International Society for Optics and Photonics, Orlando, FL, USA, 20–25 April 1997; pp. 182–193.
13. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422.
14. Cai, J.; Huang, P.; Zhang, B.; Wang, D. A TSR visual servoing system based on a novel dynamic template matching method. *Sensors* **2015**, *15*, 32152–32167.

15. Chen, L.; Huang, P.; Cai, J.; Meng, Z.; Liu, Z. A non-cooperative target grasping position prediction model for tethered space robot. *Aerosp. Sci. Technol.* **2016**, *58*, 571–581.
16. Huang, P.; Chen, L.; Zhang, B.; Meng, Z.; Liu, Z. Autonomous Rendezvous and Docking with Nonfull Field of View for Tethered Space Robot. *Int. J. Aerosp. Eng.* **2017**, *2017*.
17. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer Science & Business Media: Berlin, Germany, 2010.
18. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
19. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359.
20. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395.
21. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
22. Jigalin, A.; Gurfil, P. Investigation of multiple-baseline stereovision for state estimation of unknown dynamic space targets. *Proc. Inst. Mech. Eng. Part G* **2015**, doi:10.1177/0954410015590636.
23. Lerro, D.; Bar-Shalom, Y. Tracking with debiased consistent converted measurements versus EKF. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, *29*, 1015–1022.
24. Suchomski, P. Explicit expressions for debiased statistics of 3D converted measurements. *IEEE Trans. Aerosp. Electron. Syst.* **1999**, *35*, 368–370.
25. Xiaoquan, S.; Yiyu, Z.; Bar-Shalom, Y. Unbiased converted measurements for tracking. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 1023–1027.
26. Mei, W.; Bar-Shalom, Y. Unbiased Kalman filter using converted measurements: Revisit. In Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, San Diego, CA, USA, 4 September 2009; Volume 7445, doi:10.1117/12.831218.
27. Gallup, D.; Frahm, J.M.; Mordohai, P.; Pollefeys, M. Variable baseline/resolution stereo. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
28. Oh, J.H.; Park, J.; Lee, S.H.; Lee, B.H.; Park, J.I. Error modeling of depth measurement using FIR stereo camera systems. In Proceedings of the Third International Conference on Digital Information Processing and Communications (ICDIPC2013). The Society of Digital Information and Wireless Communication, Dubai, United Arab Emirates, 30 January–1 February 2013; pp. 470–475.
29. De Groen, P.P. An introduction to total least squares. *Nieuw Arch. Wiskd.* **1996**, *14*, 237–254.
30. Forssén, P.E.; Lowe, D.G. Shape descriptors for maximally stable extremal regions. In Proceedings of the IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
31. Rosten, E.; Drummond, T. Fusing points and lines for high performance tracking. In Proceedings of the IEEE 10th International Conference on Computer Vision, Bradford, UK, 29 June–1 July 2005; Volume 2, pp. 1508–1515.
32. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision, Providence, RI, USA, 6–13 November 2011; pp. 2564–2571.