Addressing pose estimation issues for machine vision based UAV autonomous serial refuelling

G. Campa, M. R. Napolitano and M. Perhinschi
Department of Mechanical and Aerospace Engineering
West Virginia University, Morgantown
USA

M. L. Fravolini
Department of Electronics and Information
Perugia University, Perugia
Italy

L. Pollini and M. Mammarella
Department of Electrical Systems and Automation
Pisa University, Pisa
Italy

ABSTRACT

This paper describes the results of an effort on the analysis of the performance of specific 'pose estimation' algorithms within a Machine Vision-based approach for the problem of aerial refuelling for unmanned aerial vehicles. The approach assumes the availability of a camera on the unmanned aircraft for acquiring images of the refuelling tanker; also, it assumes that a number of active or passive light sources – the 'markers' – are installed at specific known locations on the tanker. A sequence of machine vision algorithms on the on-board computer of the unmanned aircraft is tasked with the processing of the images of the tanker. Specifically, detection and labeling algorithms are used to detect and identify the markers and a 'pose estimation' algorithm is used to estimate the relative position and orientation between the two aircraft.

Detailed closed-loop simulation studies have been performed to compare the performance of two 'pose estimation' algorithms within a simulation environment that was specifically developed for the study of aerial refuelling problems. Special emphasis is placed on the analysis of the required computational effort as well as on the accuracy and the error propagation characteristics of the two methods. The general trade offs involved in the selection of the pose estimation algorithm are discussed. Finally, simulation results are presented and analysed.

NOMENCLATURE

3DW	Three Dimensional Window
AR	aerial refuelling
CG	centre of gravity
DAL	detection and labeling
FE	feature extraction
GLSDC	Gaussian least squares differential correlation
GPS	global positioning system
LHM	Lu, Hager and Mjolsness
MV	machine vision
PE	pose estimation
PSD	power spectral density
RF	reference frame
UAV	unmanned aerial vehicle
VR	virtual reality
VRT	virtual reality toolbox®
WGN	white Gaussian noise

1.0 INTRODUCTION

One of the biggest current limitations of unmanned aerial vehicles (UAVs) is their lack of aerial refuelling (AR) capabilities. Currently

Paper No. 3110. Manuscript received 1 June 2006, revised 22 March 2007 accepted 27 April 2007.

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press

two hardware configurations are used for the aerial refuelling of manned aircraft. The first one is used by the US Air Force and features a refuelling boom – manoeuvered by a human boom operator – which connects with the fuel receptacle of the aircraft to be refueled. This configuration is known as 'refuelling boom' system. The second configuration is used by the US Navy and features instead a flexible hose equipped with an aerodynamically stabilised perforated cone. This last configuration is known as the 'Probe and Drogue' system. The effort described in this paper is mostly relative to the US Air Force refuelling boom system and has the general goal of extending the use of this system to the refuelling of UAVs[1,2]. For this purpose a key issue is represented by the need of accurate measurement of the 'tanker-UAV' relative position and orientation from a 'pre-contact' to a 'contact' position and during the refuelling. In fact, although sensors based on laser, infrared radar, and GPS technologies are suitable for autonomous docking[3], there might be limitations associated with their use. For example, the use of UAV GPS signals might not always be possible since the GPS signals may be shadowed or distorted by the tanker airframe. Therefore, the use of machine vision (MV) technology has been proposed in addition – or as an alternative – to these technologies[4]. Modelling and control issues related to the introduction of a MV position sensing system were discussed in[5-7], for the 'probe and drogue' refuelling system. Specific algorithms suitable for aerospace MV systems were also discussed in[8] – within the context of close proximity operations of aerospace vehicles – and in[9] – within the context of autonomous navigation and landing of aircraft.

Control issues related to the boom-based refuelling approach were investigated in[10], while the development of a GPS-based simulation environment – including the modelling of the boom operator – to be used within the air force research laboratory (AFRL), was discussed in Refs 11 and 12.

A MV-based system for the estimation of the UAV-Tanker relative position and orientation assumes the availability of a digital camera – installed on the UAV – providing the images of the tanker. These images are then processed by various algorithms to solve a pose estimation problem, leading to real-time estimates of the relative position and orientation vectors. These vectors are then used for the purpose of guiding the UAV from a 'pre-contact' to a 'contact' position. Once the UAV reaches the contact position, the boom operator takes over and manually proceeds to the docking of the refuelling boom followed by the actual refuelling operation.

The main contribution of this paper is a detailed comparison of the performance of the most commonly used pose estimation algorithms within a closed-loop simulation.

The paper is organised as follows. The AR problem and the simulation environment are described in the initial sections. The point matching problem and the implementation of a novel labeling algorithm are briefly presented next. Then, two different pose estimation algorithms are introduced and extensively compared in terms of speed, accuracy, robustness, and error propagation characteristics.

2.0 THE MV-BASED AAR PROBLEM

This section formally introduces the geometric formulation of the AR problem and the required assumptions.

2.1 Reference frames and notation

The following reference frames (RFs) are required for the study of the AR problem:

- ERF, or *E*: earth-fixed reference frame.
- TRF or *T*: body-fixed tanker reference frame located at the tanker centre of gravity (CG).
- URF or *U*: is the body-fixed UAV reference frame located at the UAV CG.
- CRF or *C*: is the body-fixed UAV camera reference frame.

Within this study, geometric points are expressed using homogeneous (4D) co-ordinates and are denoted with a capital letter and a left superscript indicating the reference frame in which the point is expressed. For example, a point P expressed in the F reference frame, has co-ordinates ${}^{F}P = [x,y,z,1]^{T}$, (where the right 'T' superscript indicates transposition). Vectors are defined as difference between points; therefore, their 4th co-ordinate is always '0'. Also, vectors are denoted by two uppercase letters, indicating the two points at the extremes of the vector; for example, ${}^{E}BR = {}^{E}B - {}^{Ev}R$ is the vector from the point R to the point B, expressed in the Earth Reference Frame. Transformation matrices are (4 × 4) matrices that transform points and vectors expressed in a first reference frame to points and vectors expressed in a second reference frame. They are usually denoted with a capital T, with a right subscript indicating the 'first' reference frame and a left superscript indicating the 'second' reference frame. For example the matrix represents the homogeneous transformation matrix ${}^{E}T_{T}$ that transforms a vector/point expressed in TRF to a vector/point expressed in ERF.

2.2 Geometric formulation of the AAR problem

The objective is to guide the UAV such that its fuel receptacle (identified as point R) is transferred to the centre of a 3-dimensional window (3DW) under the tanker (referred as point B). Once the UAV fuel receptacle reaches this 3DW and remains within it, the boom operator takes control of the refuelling operations. It should be emphasised that point B is fixed within the TRF, and that the dimensions of the 3DW are one of the design parameters. It is also assumed that the tanker and the UAV share a low-bandwidth short-range data communication link during the docking manoeuvre. Furthermore, it is assumed that the UAV is equipped with a digital camera along with an on-board digital computer. Finally, the 2D image plane of the MV is defined as the 'y-z' plane of the CRF.

2.3 Receptacle-3DW-centre vector

The reliability of the AAR docking manoeuvre is based on the accuracy of the measurement of the vector ${}^{T}RB$, which is the distance between the UAV fuel receptacle and the centre of the 3D refuelling window, expressed in TRF:

$$ {}^{T}RB = {}^{T}B - {}^{T}R - {}^{T}T_{U}\,{}^{U}R \qquad \ldots (1) $$

In the above formula, ${}^{T}T_{U}$ represent the UAV position and orientation with respect to the tanker. Since the fuel receptacle and the 3DW centre are located at fixed and known positions with respect to the centres of gravity of the UAV and tanker respectively, the vectors ${}^{U}R$ – receptacle co-ordinated in UAV body frame – and ${}^{T}B$ – box centre co-ordinates in tanker frame – are known and constant. The matrix ${}^{T}T_{U}$ can be calculated using the relationship ${}^{T}T_{U} = ({}^{U}T_{C}, {}^{C}T_{T})$ where the matrix ${}^{U}T_{C}$ expresses the position and attitude of the camera with respect to the UAV. Therefore the only unknown for the calculation of the vector ${}^{T}RB$ is the transformation matrix ${}^{C}T_{T}$, which represents the tanker position and orientation with respect to the UAV camera. This matrix can be evaluated either 'directly', that is using the relative position and orientation information provided by the MV system, or 'indirectly', that is by using the formula , where the matrices ${}^{E}T_{U}$ – expressing UAV position and orientation in earth frame – and ${}^{E}T_{T}$ – expressing tanker position and orientation in earth frame- can in turn be evaluated using data from the position and attitude sensors of both tanker and UAV.

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press

3.0 THE AAR SIMULATION ENVIRONMENT

The AAR simulation scheme was developed in Simulink®[13]. The mathematical models of the tanker and UAV aircraft were developed using conventional non-linear state space modelling approaches outlined in Ref. 14. Specifically the UAV was modeled with the parameters of the ICE-101 aircraft[15] and the tanker was modeled with the parameters of a KC-135 aircraft[16]. First order dynamic models were used for the actuators dynamics with typical values for aircraft of similar size and/or weight. Both the tanker and UAV feature a typical autopilot system designed using a conventional LQR approach. The simulation code also includes detailed modelling for the UAV sensors, the elastic behaviour of the boom, the wake effects induced by the tanker on the UAV, as well as atmospheric turbulence models acting on both the tanker and the UAV[13]. The simulation outputs were linked to a virtual reality toolbox® (VRT) interface to provide the scenery associated with the AAR manoeuvres. The interface allows the positions of the simulated objects (UAV and tanker) to drive the position and orientation of the corresponding objects in a virtual world. Within this virtual world, optical markers – modeled as red spheres with a 10cm radius – were installed on the KC-135 tanker aircraft at specific locations, as discussed in[5].

4.0 DETECTION AND LABELING ALGORITHM

The images of the tanker – as seen from a virtual camera placed on the UAV – are continuously acquired from the virtual world and processed throughout the simulation. Figure 1 shows a typical image acquired by the virtual camera. The extraction of the camera-plane 2D co-ordinates $[u_j, v_j]$ of the markers is performed using standard filtering techniques[17]. The following task is to entails associating the 2D camera-plane co-ordinates of each detected marker with the correct physical marker on the tanker aircraft, whose position in TRF (3D co-ordinates) is known. In other words, a set of detected markers $[u_j, v_j]$ has to be associated to a subset of estimated markers positions.

4.1 Projection equations

The subset $[\hat{u}_j, \hat{v}_j]$ is simply a projection in the camera plane of the markers $P_{(j)}$, where the projection is performed using the standard 'pin-hole' model[4,18]. Specifically, according to the 'pin-hole' projection model, given a marker 'j' with co-ordinates ${}^{c}P_{(j)} = [{}^{c}x_j, {}^{c}y_j, {}^{c}z_j, 1]^T$ in CRF, its projection into the image plane can be calculated using the following standard equation:

$$\begin{bmatrix} \hat{u}_j \\ \hat{v}_j \end{bmatrix} = \frac{f}{{}^{c}x_{p,j}} \begin{bmatrix} {}^{c}y_{p,j} \\ {}^{c}z_{p,j} \end{bmatrix} = g\left(f, {}^{C}T_T(X) \cdot {}^{T}P_{(j)}\right) \quad \ldots (2)$$

where f is the camera focal length, ${}^{T}P_{(j)}$ are the components of the marker $P_{(j)}$ in TRF, which are fixed and known 'a priori', and ${}^{C}T_T(X)$ is the transformation matrix between the camera and tanker reference frames, which is a function of the current position and orientation vector X:

$$X = [{}^{c}x_T, {}^{c}y_T, {}^{c}z_T, {}^{c}\psi_T, {}^{c}\theta_T, {}^{c}\varphi_T]^T \quad \ldots (3)$$

For labeling purposes X is assumed to be known. In fact, the camera-tanker distance – i.e. the first three elements of X – can be provided by the GPS measurements on both the tanker and UAV, if adequate GPS coverage is available. In particular, the data acquired from the tanker GPS can be broadcasted to the UAV control system using the short-range radio link available for *AR* purposes. Alternatively, the MV-based estimation of the camera-tanker distance at previous time instants can be used as a good approximation of the current distance (assuming a sufficiently fast sampling rate for the MV system). The relative orientation between camera and tanker – that is, the last three elements of X – can be obtained from the yaw, pitch, and roll angle measurements of both UAV and tanker, which are provided by conventional gyros. The modelling for these sensors also included the addition of a white noise with power spectral density of 10-9dB. Alternatively, as also done for the distance calculation, if the sampling rate of the MV system is adequate, the last MV estimation of the camera-tanker relative orientation can be used as a good approximation of the current orientation.

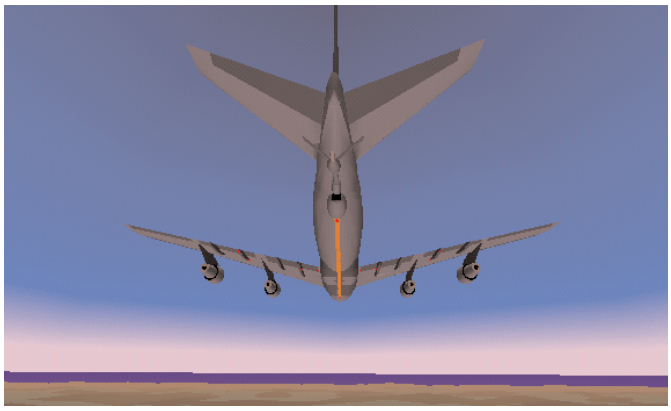


Figure 1. Typical image acquired by the camera in the VR world.

4.2 The 'points matching' problem

Once the 'projections' subset $[\hat{u}_j, \hat{v}_j]$ is available, the problem of relating the 2D points extracted from the camera measurements to the actual markers on the tanker can be formalised in terms of matching the set of points $\{p_1, p_2, \ldots, p_m\}$ – where $p_j = [u_j, v_j]$ is the generic 'to be matched' point from the camera – to the set of points $\{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n\}$ – where $\hat{p}_j = [\hat{u}_j, \hat{v}_j]$ is the generic point obtained by projecting the markers in the camera plane through Equation (2). Since the two data sets represent the 2D projections of the same points at the same time instant on the same plane, a high degree of correlation between the two sets is expected. In the ideal case – that is assuming the availability of immediate and perfect measurements – the corresponding points would be exactly super-imposed, resulting in a trivial matching process. However, in the presence of different sources of system and measurement noise, a matching problem has to be defined and solved.

A detailed technical literature describes a number of robust matching techniques between point sets[19]. Usually, the degree of similarity between two data sets is defined in terms of a cost function or a distance function derived on general principles as geometric proximity, rigidity, and exclusion[20]. A definition of the point-matching problem as an assignment problem along with a comparison of different matching algorithms was performed by some of the authors in a previous effort[21,22]. The algorithm implemented within this effort solves the problem using a heuristic 'mutual nearest point' procedure[23], which uses mainly differences in point positions and is implemented as a level-2 Simulink S-Function written in *C*, for real-time implementation purposes.

5.0 POSE ESTIMATION ALGORITHMS

Following the solution of the labeling problem, the information in the set of points $\{p_1, p_2, \ldots, p_m\}$ must be used to derive the rigid transformation relating the CRF to the TRF[24,25]. The literature offers a wide range of alternatives to solve this typical 2D to 3D correspondence problem. Many of the available methods are

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press

sensitive to parameters like the number of detected point or the length of the image sequence[26]. Algorithms that rely on the estimation of the kinematics and structure of the rigid object for solving a recursive pose estimation problem have been presented in Refs 27 and 28. Within this study, two algorithms to solve the more specific problem of estimating relative position and orientation from a set of 2D to 3D point correspondences within a single image were implemented and compared.

5.1 The GLSDC algorithm

The Gaussian least squares differential correlation (GLSDC) algorithm is based on the application of the Gauss-Newton method for the minimisation of a non-linear cost function expressing the difference between estimated and detected markers positions[24]. This algorithm was selected because it represents a class of algorithms which is still very widely used in photogrammetry. The GLSDC algorithm has a simple and understood structure; additionally, it has actually been already used to solve pose estimation problems within an autonomous aerial refuelling setting, as described in Ref. 4. Furthermore, its simple structure allows a straightforward real time implementation, which, in turn, allowed the authors to perform an actual experiment using off the shelf hardware and a scaled B777 model. Details of the experiments are described in Ref. 31. Within the GLSDC algorithm, at every sample time k, the matrix ${}^{C}T_{T}$ is expressed as a function of an estimate $\bar{X}(k)$ of the unknown vector $X(k)$:

$$\bar{X}(k) = [\,{}^{c}\bar{x}_{T}\,,\,{}^{c}\bar{y}_{T}\,,\,{}^{c}\bar{z}_{T}\,,\,{}^{c}\bar{\psi}_{T}\,,\,{}^{c}\bar{\theta}_{T}\,,\,{}^{c}\bar{\varphi}_{T}\,]^{T} \quad \ldots (4)$$

Using $\bar{X}(k)$ to project of the marker 'j' in the camera plane yields the following 2D co-ordinates:

$$\begin{bmatrix}\bar{u}_j\\ \bar{v}_j\end{bmatrix} = \frac{f}{{}^{c}\bar{x}_{p,j}}\begin{bmatrix}{}^{c}\bar{y}_{p,j}\\ {}^{c}\bar{z}_{p,j}\end{bmatrix} = g\left(f, {}^{C}T_{T}(\bar{X}(k))\cdot {}^{T}P_{(j)}\right) \quad \ldots (5)$$

By rearranging the co-ordinates of all the projected markers, the following vector is obtained

$$G(\bar{X}(k)) = \left[\bar{u}_1, \bar{v}_1, \ldots\ldots, \bar{u}_m, \bar{v}_m\right] \quad \ldots (6)$$

At this point, the following MV estimation error can be defined at the time k:

$$\Delta G(k) = G_{DAL}(k) - G(\bar{X}(k)) \quad \ldots (7)$$

where $G_{DAL}(k)$ contains the co-ordinates of the detected and labeled points extracted from the camera:

$$G_{DAL}(k) = \left[u_1, v_1, \ldots\ldots, u_m, v_m\right] \quad \ldots (8)$$

the GLSDC algorithm iteratively refines the initial value of $\bar{X}(k)$ by repeating the following steps for a number of iterations (with index i):

$$\bar{X}_{i+1}(k) = \bar{X}_i(k) + R_i^{-1}(k)A_i^{T}(k)W(k)\Delta G_i(k) \quad \ldots (9)$$

where

$$R_i(k) = A_i^{T}(k)W(k)A_i(k) \quad \ldots (10)$$

$$A_i(k) = \left.\frac{\partial G_i(k)}{\partial X}\right|_{X=\bar{X}_i(k)} \quad \ldots (11)$$

and $W(k)$ is usually set to the (2m × 2m) covariance matrix of the estimation error.

The initial guess $\bar{X}_0(k)$ at k is the final estimation at the previous sample time k-1. The basic algorithm outlined in Equations (9)-(11), was designed to work with a fixed number of m markers. Simple modifications have been introduced for dealing with a time-varying number of markers. Specifically, at each time step, the nominal markers that are not visible are removed from the estimation process. This implies that, at each time step, Eq. (6) is modified such that it includes only the appropriate number of rows; thus, the dimensions and the values of A and W in Equations (9)-(11) are adjusted accordingly.

5.1.1 The LHM algorithm

In Ref. 29, Lu, Hager and Mjolsness (LHM) proposed an algorithm based on the minimisation of an object-space collinearity error. Specifically, given the 'observed, detected and labeled' point 'j' on the camera plane at the time instant k with co-ordinates $[u_j,v_j]$, let $h_j(k)$ be:

$$h_j(k) = [u_j\ v_j\ 1]^{T} \quad \ldots (12)$$

Then, an 'object-space collinearity error' vector e_j, at the time instant k, can be defined as follows:

$$e_j(k) = (I - V_j(k))\,{}^{C}T_{T}(\bar{X}(k))\,{}^{T}P_{(j)} \quad \ldots (13)$$

where

$$V_j(k) = \begin{bmatrix}\dfrac{h_j(k)h_j^{T}(k)}{h_j^{T}(k)h_j(k)} & 0\\ 0 & 1\end{bmatrix} \quad \ldots (14)$$

The *PE* problem is then formulated as the problem of minimising the sum of the squared errors:

$$E(\bar{X}(k)) = \sum_{j=1}^{m}\left\|e_j(k)\right\|^{2} \quad \ldots (15)$$

The algorithm operates by iteratively improving an estimate of the rotation portion of the pose and then estimates the associated translation when a satisfying estimate of the rotation is found. In practice, this is accomplished by exploiting the structure of the collinearity equations:

$$\begin{bmatrix}\dfrac{\hat{h}_j\hat{h}_j^{T}}{\hat{h}_j^{T}\hat{h}_j} - 1 & 0\\ 0 & 0\end{bmatrix}{}^{C}T_{T}\,{}^{T}P_{(j)} = 0 \quad \ldots (16)$$

where

$$\hat{h}_j = \begin{bmatrix}\hat{u}_j & \hat{v}_j & 1\end{bmatrix}^{T} \quad \ldots (17)$$

and $[\hat{u}_j,\hat{v}_j]$ is the projection in the camera plane of the point $TP(j)$. The LHM algorithm represents the state of the art in the solution of the pose estimation problem from a single set of 2D-3D point correspondences. It has been shown[29] that the LHM algorithm is globally convergent in the sense that it always converges to the rotation matrix that minimises the collinearity error for any set of observed point and any initial rotation matrix. Furthermore, empirical results suggest that the algorithm is also very efficient and usually converges within five to ten iterations starting from any range of initial conditions.

6.0 RESULTS OF THE SIMULATION STUDIES

The following performance criteria were introduced for a detailed comparison of the two algorithms:

1. Speed

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press

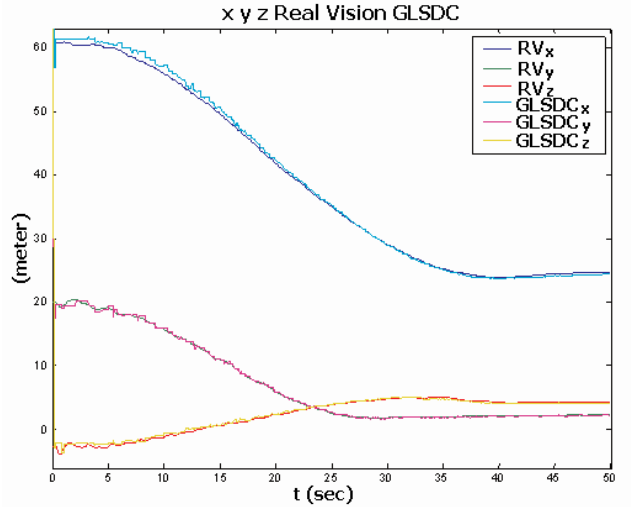


Figure 2. 'Real' *x y z vs x y z* estimates from the GLSDC algorithm.

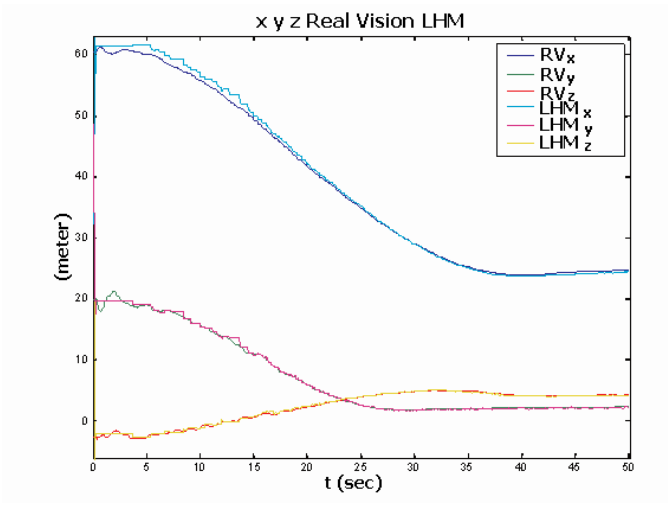


Figure 3. 'Real' *x y z vs x y z* estimates from the LHM algorithm.

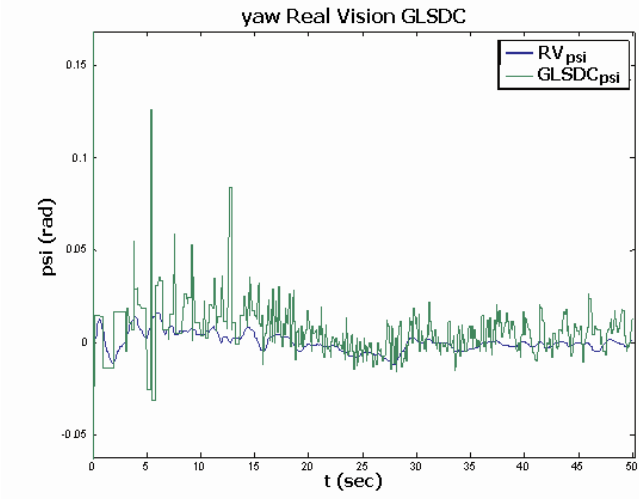


Figure 4. 'Real' yaw angle *vs* estimate from the GLSDC algorithm.

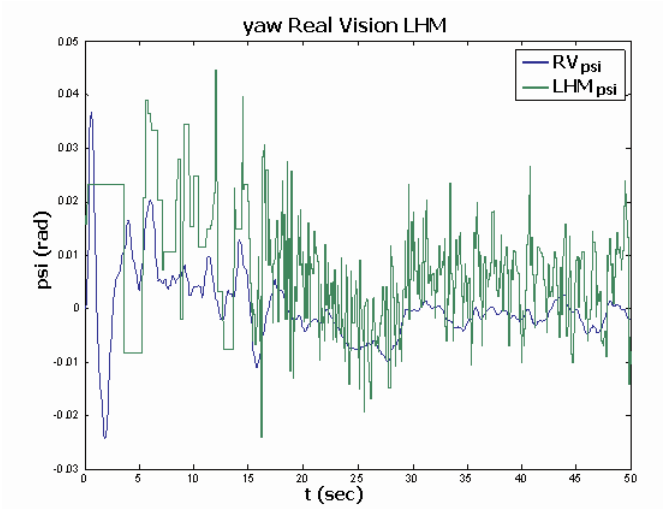


Figure 5. 'Real' yaw angle *vs*. estimate from the LHM algorithm

2. Accuracy
3. Robustness
4. Tracking error

6.1 Speed performance

A Pentium 4, 2·53GHz laptop with 448Mbytes of RAM was used for this analysis. The speed performance was measured with the Simulink® 'profiler' tool, which provides the running time in seconds for each called function and sub-function. The simulation lasted 40 seconds; and the MV system used a sampling time of 0·1 sec. On average, the GLSDC and LHM algorithms required $5{\cdot}3*10^{-3}$ sec and $20{\cdot}1*10^{-3}$ sec per simulation step, respectively. Thus the LHM algorithm is approximately four times slower than the GLSDC.

6.1.1 Accuracy (difference between true and estimated ${}^{C}T_{T}$ values)

For the purpose of this analysis, 'true values' are defined as the distance and orientation of the tanker in camera frame read at each simulation step from the linear and angular position (simulated) sensors. The 'estimated values' are instead the distances and orientations of the tanker in camera frame provided directly by the two pose estimation algorithms. It should be emphasised that both the true and estimated values are slightly dependent on the particular UAV trajectory, which is in turn dependent on the linear and angular position estimations provided by the particular PE algorithm that is being used within the control loop.

An analysis of the results highlights that the LHM needs a minimum of five markers while the GLSDC requires at least fourt markers. However, the GLSDC performance strongly depends on the accuracy of the initial conditions. Table 1 shows the results from t_1 = 15sec until t_2 = 50sec.

Table 1
RMS values of the error for the PE algorithms between t_1 = 15sec and t_2 = 50 sec

	X	*Y*	*Z*	Roll	Pitch	Yaw
GLSDC	0·3254	0·0962	0·1679	0·0090	0·0150	0·0098

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press

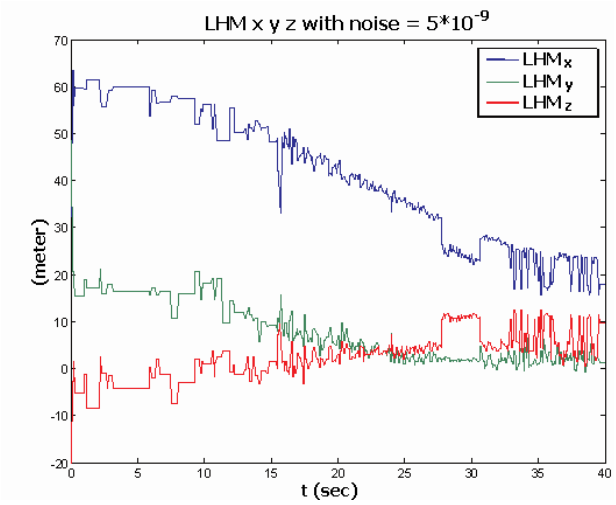


Figure 6. LHM behaviour with $5*10^{-9}$ noise power.

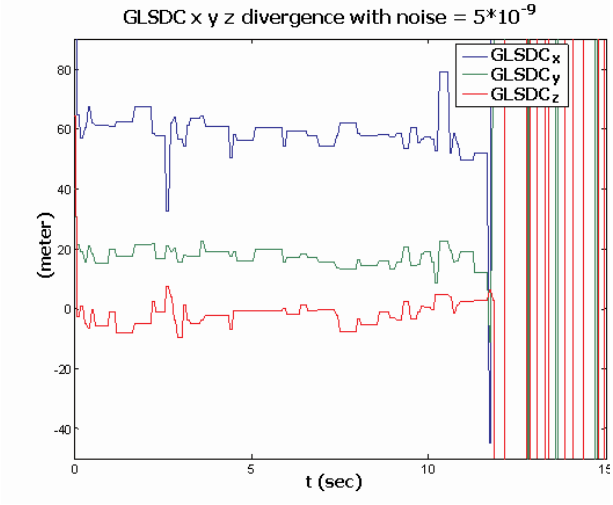


Figure 7. GLDSC behaviour with $5*10^{-9}$ noise power.

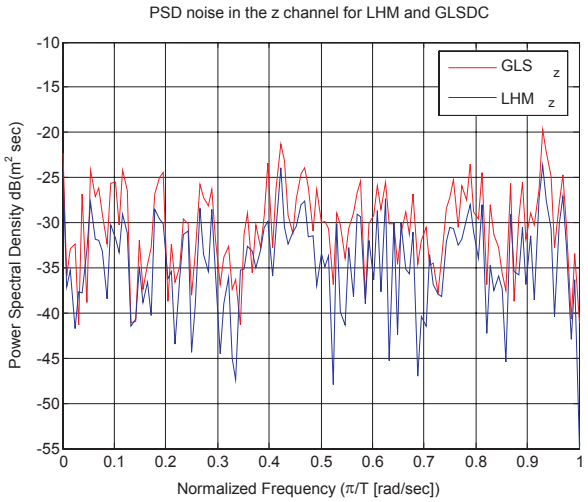


Figure 8. PSD of GLSDCZ and LHMZ with noise $1*10^{-9}$.

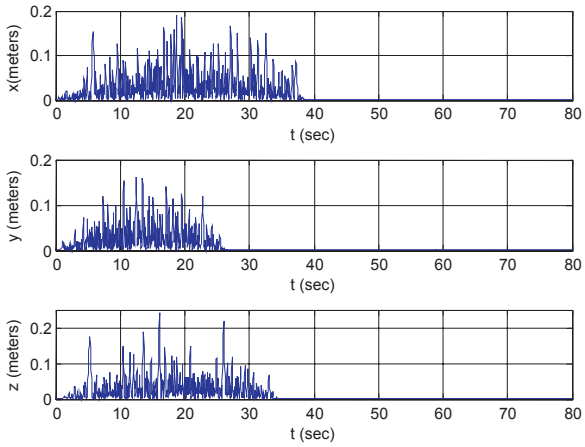


Figure 9. Tracking error (Using the LHM algorithm).

LHM	0·3364	0·0963	0·1367	0·0109	0·0125	0·0104

Overall, it appears that the GLSDC and LHM algorithms provide similar levels of accuracy. Figures 2 to 5 show the linear and angular position estimations from the two algorithms.

6.1.2 Robustness

This analysis was performed in terms of robustness with respect to the following:

- noise addition in the markers position (with correct labeling).
- incorrect performance of the labeling algorithm.
- errors in initial conditions.
- input noise.

Noise addition in the markers position with correct labeling

This analysis was performed with the MV algorithms in open-loop mode, that is, the docking control laws used values from sensors and GPS to perform the docking manoeuvre. Different levels of noise were added to the correctly labeled 2D markers positions. Specifically, the selected noise was a band limited white noise with correlation time t_c = 0·05sec, the value of the power spectral density (PSD) in the bandwidth of interest was entered trough the 'Noise Power' parameter. Different values of the PSD were evaluated, starting form 0 to $5*10^{-9}$ m^2 with an interval of $1*10^{-9}$ m^2. Up to a value of $4*10^{-9}$ m^2, both algorithms performed reasonably well. For a PSD of $5*10^{-9}$ m^2 the LHM performance started to deteriorate, as shown in Fig. 6.

However, in order for this performance degradation to take place, the input noise has to act in a situation in which the number of detected markers is limited. In fact, better performance can promptly be recovered whenever the number of detected markers increases and/or the noise power levels decrease. Conversely, it was observed that for a PSD of $5*10^{-9}$ m^2 the GLSDC performance tend to degrade abruptly, leading the algorithm outside of its stability region, as shown in Fig. 7.

In this event the algorithm is not able to recover an acceptable performance, independently on the number of detected markers. On the other hand, whenever the number of detected markers is greater than five, both algorithms provide similar desirable

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press

performance.

Robustness to labelling errors

In this study, incorrect performance of the labeling algorithms were simulated by inverting the labeling of two markers at two simulation steps, specifically for t = 20 sec and t = 20·1sec. During these time instants the markers #8 and #7 were exchanged and provided as inputs to the pose estimation algorithms. The analysis shows that the GLSDC is substantially more sensitive to labeling errors than the LHM algorithm. For example, the x estimated by the GLSDC drops from 42·8m to 12·6m against the 33·5m provided by the LHM; a similar behaviour was observed for the other estimated variables. An even more important conclusion is that the GLSDC algorithm requires a considerably longer time to return to the 'nominal' performance than the LHM algorithm. This conclusion is consistent with the previous results on the noise robustness of the two algorithms.

Robustness to errors in initial conditions

The purpose of this study was to evaluate the performance of the GLSDC and LHM algorithms under a broad range of errors in the initial linear and angular position of the tanker in camera frame. To perform the test, the number of detected markers was held constant at 5 for both algorithms; a larger convergence area is of course expected whenever additional visible markers are detected. The exact initial conditions were set to be xo= [x, y, z, psi, theta, phi] = [60·5 20 –2·5 0 0·467 0]. The results, which are summarised in Table 2, show that the GLSDC algorithm has a limited convergence area while the LHM algorithm performs very well for any range of erroneous initial conditions.

Furthermore, within its convergence area, the GLSDC shows a larger settling time before providing reasonably accurate estimates. On the other side, while outside its convergence area, the GLSDC algorithm either diverges or provides inaccurate estimates.

Error propagation analysis

In this study the noise propagation behaviour of the two algorithms was investigated. Specifically, a white Gaussian noise (WGN) was added to the position (x, y) of one single marker. This noise propagates through the pose estimation algorithm, resulting in a noisy output. It was observed that the output noise retains the same properties of the input noise, that is, Gaussian with zero mean, and with a PSD proportional to the power of the input noise. It is interesting to note that, while the GLSDC and LHM algorithms propagated the errors in a similar manner for the variables x and y, the GLSDC algorithm amplified the input noise more than the LHM algorithm along the z channel, as shown in Fig. 8. The properties of the output noise suggest that a Kalman-Filter based approach could be highly effective in handling the measurements provided by the MV system. Detailed descriptions of the output signal properties of the MV system are provided in Ref. 30.

6.1.3 Tracking error analysis

Table 2
Convergence region for the initial condition

Translation vector		Yaw angle (psi)	
GLSDC interval from exact initial condition	LHM interval from exact initial condition	GLSDC interval from exact initial condition	LHM interval from exact initial condition
[–44·7 65·4]	[–∞ +∞]	[–1·74 3·06] rad =	[0·2π]

Table 3
Mean tracking error with *GLSDC* and *LHM*

	GLSDC (metre)	LHM (metre)
x	$1{\cdot}3831*10^{-2}$	$1{\cdot}4180*10^{-2}$
y	$1{\cdot}0154*10^{-2}$	$0{\cdot}9996*10^{-2}$
z	$1{\cdot}2835*10^{-2}$	$1{\cdot}3674*10^{-2}$

As a final study, the closed-loop tracking error (that is the difference between the actual docking path and the nominal docking path) has been evaluated for both LHM and GLSDC algorithms. Since the level of accuracy for the two algorithms is comparable, it was not expected for the tracking error to be influenced by the performance of the pose estimation algorithm.

Figure 9 shows the tracking error along the three axes (in the LHM case) while Table 3 shows the mean of the tracking error for the two algorithms.

7.0 CONCLUSIONS

This paper described different approaches for the Pose Estimation problem within MV algorithms that were implemented and tested within a simulation environment specifically developed for the study of the MV-based AAR problem. In particular, the attention focused on the analysis of the performance of two widely used pose estimation algorithms – the GLSDC and the LHM algorithm – in terms of accuracy and robustness. The results from this comparison indicate that the accuracy of the two algorithms is substantially similar; however, the LHM algorithm provides a substantially higher level of robustness at the expense of a larger required computational effort. Therefore the LHM is to be preferred whenever its additional computational requirements are not an issue.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and useful suggestions.

REFERENCES

1. NALEPKA, J.P. and HINCHMAN, J.L. Automated aerial refuelling: extending the effectiveness of unmanned air vehicles, AIAA Modelling and Simulation Technologies Conference and Exhibit, 15-18 August 2005, San Francisco, CA, USA.
2. ZHIPU JIN, Z., SHIMA, T. and SCHUMACHER, C.J. Optimal scheduling for refuelling multiple autonomous aerial vehicles IEEE Transactions on robotics, August 2006, **22**, (4).
3. MONDA, E.W., LIGHTSEY, E.G. and KEY, K. An Investigation of GPS pseudolite based relative navigation, *Advances in the Astronautical Sciences*, 2004, **3**, (529).
4. VALASEK, J., GUNNAM, K., KIMMETT, J., TANDALE, M.D., JUNKINS, J.L. and HUGHES, D. Vision-based sensor and navigation system for autonomous air refuelling, *J Guidance, Control, and Dynamic*s, September – October 2005, **28**, (5).
5. FRAVOLINI, M.L., FICOLA, A., CAMPA, G., NAPOLITANO M.R. and SEANOR, B., Modelling and control issues for autonomous aerial refuelling for UAVs using a probe-drogue refuelling system, *J Aerospace Science Technology*, 2004, **8**, (7), pp 611-618.
6. POLLINI, L., INNOCENTI, M. and MATI, R. Vision algorithms for formation flight and aerial refuelling with optimal marker labeling, AIAA Modelling and Simulation Technologies Conference and Exhibit, 15-18 August 2005, San Francisco, CA, USA.
7. HERRNBERGER, M., SACHS, G., HOLZAPFEL, F., TOSTMANN, W. and WEIXLER, E. Simulation analysis of autonomous aerial refuelling procedures AIAA Guidance, Navigation, and Control Conference and Exhibit, 15-18 August 2005, San Francisco, CA, USA.
8. KELSEY, J.M., BYRNE, J., COSGROVE, M., SEEREERAM, S. and MEHRA, R.K. Vision-based relative pose estimation for autonomous rendezvous

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press

and docking, 2006 IEEE Aerospace Conference, 4-11 March 2006, Big Sky, MT, USA.
9. CHATTERJI, G.B., MENON, P.K. and SRIDHAR, B. GPS/Machine vision navigation system for aircraft, *IEEE Transactions on Aerospace and Electronic Systems*, July 1997, **33**, (3), pp 1012-1025.
10. ROSS, S.M., PACHTER, M., JACQUES, D.R., KISH, B.A. and MILLMAN, D.R. Autonomous aerial refuelling based on the tanker reference frame, 2006 IEEE Aerospace Conference, 4-11 March 2006, Big Sky, MT, USA.
11. NGUYEN, B.T. and LIN, L.T. The use of flight simulation and flight testing in the automated aerial refuelling program, AIAA Modelling and Simulation Technologies Conference and Exhibit, 15-18 August 2005, San Francisco, CA, USA.
12. BURNS S.R., CLARK, C.S. and EWART, R. The automated aerial refuelling simulation at the AVTAS Laboratory, AIAA Modelling and Simulation Technologies Conference and Exhibit, 15-18 August 2005, San Francisco, CA, USA.
13. CAMPA G., NAPOLITANO M.R. and FRAVOLINI, M.L. A simulation environment for machine vision based aerial refuelling for UAVs, Submitted to: IEEE Transaction On Aerospace and Electronic Systems, April 2006, accepted with minor revisions November 2006.
14. STEVENS, B.L. and LEWIS, F.L. *Aircraft Control and Simulation*, John Wiley & Sons, 1987, New York, USA.
15. ADDINGTON, G.A. and MYATT, J.H. Control-surface deflection effects on the innovative control effectors (ICE 101) design, Air Force Report, AFRL-VA-WP-TR-2000-3027, June 2000.
16. Air Force Online Database 2005.: http://www.af.mil/factsheets.
17. Image Processing Toolbox for use with MATLAB: User's Guide September 2003, The Mathworks Inc.
18. HUTCHINSON. S., HAGER, G. and CORKE, P. A tutorial on visual servo control, *IEEE Transactions on Robotics and Automation*, 1996, **12**, (5), pp 651-670.
19. UMEYAMA, S. Parameterised point pattern matching and its application to recognition of object families, *IEEE Transactions on pattern analysis and machine intelligence*, **15**, (2), 1993, pp 136-144.
20. PLA, F. and MARCHANT, J.A. Matching feature points in image sequences through a region-based method, *Computer vision and image understanding*, 1997, **66**, (3), pp 271-285.
21. FRAVOLINI, M.L., CAMPA, G., NAPOLITANO, M.R. and FICOLA, A. Evaluation of machine vision algorithms for autonomous aerial refuelling for unmanned aerial vehicles, Submitted to: *AIAA J Aerospace Computing, Information and Communication*, April 2005.
22. CAMPA, G., MAMMARELLA, M., NAPOLITANO, M.R., FRAVOLINI, M.L., POLLINI, L. and STOLARIK, B. A comparison of pose estimation algorithms for machine vision based aerial refuelling for UAV, Mediterranean Control Conference 2006, 28-30 June 2006, Ancona, Italy.
23. FRAVOLINI, M.L., BRUNORI, V., FICOLA, A., LA CAVA, M. and CAMPA, G. Feature matching algorithms for machine vision based autonomous arial refuelling, Mediterranean control conference 2006, 28-30 June 2006, Ancona, Italy.
24. HARALICK, R.M. *et al*, Pose estimation from corresponding point data, *IEEE Transactions on systems, man, and cybernetics*, 1989, **19**, (6), pp 1426-1446.
25. WILSON, W. Visual servo control of robots using Kalman filter estimates of robot pose relative to work-pieces, Visual Servoing, HASHIMOTO, K. (Ed), *World Scientific*, 1994, pp 71-104.
26. AGGARWAL, J. and NANDHAKUMAR, N. On the computing of motion from sequences of images – a review, Proceeding of the IEEE, 1988, **76**, (8), pp 917 – 935.
27. BROIDA, T. and CHELLAPPA, R. Estimating the kinematics and structure of a rigid object from a sequence of monocular images, *IEEE transaction on pattern analysis and machine intelligence*, 1991, **13**, (6), pp 497-513.
28. SOATTO, S. and PERONA, P. Reducing structure from motion: a general framework for dynamic vision, *Part 1: modelling, IEEE transaction on pattern analysis and machine intelligence*, 1998, **20**, (9), pp 993-942,.
29. LU, C.P., HAGER, G.D. and MJOLSNESS, E. Fast and globally convergent pose estimation from video images, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 2000, **22**, (6), pp 610-622.
30. MAMMARELLA, M. Addressing pose estimation issues for application of machine vision to UAV autonomous aerial refuelling, (2005), Master Thesis, DSEA, Pisa http://etd.adm.unipi.it/theses/available/etd-06182005182620/unrestricted/Tesi1_12.pdf
31. DELL' AQUILA, R., CAMPA, G., NAPOLITANO, M. R. and MAMMARELLA, M. Real-time machine-vision-based position sensing system for UAV aerial refuelling, *J Real-Time Image Processing*, **1**, (3), pp 213-224 April 2007.

https://doi.org/10.1017/S0001924000004644 Published online by Cambridge University Press