Simultaneous Localization and Mapping with Monocamera: Detection of Landing Zones

Nuno M. B. Moutinho

ISR - Instituto de Sistemas e Robótica; Departamento de Engenharia Aerospacial; IST - Instituto Superior Técnico, Lisbon, November of 2009

***Abstract*—The development of autonomous systems has been growing throughout the years. The capacity systems have to, simultaneously, locate and map unknown environments, without any kind of human intervention, empowered the unknown's exploration evolution to levels never considered before. It is therefore necessary for this kind of system to have capabilities of action and decision in the required mission phases for which it was designed. The development of tools that can interact in the decision process of certain situations, constitutes an advance, bringing the systems closer to the desired autonomy level. However, complete autonomy is yet to be reached.**

The work presented on this thesis focus on the development of a simultaneous tracking and mapping system, which employs only one camera as a sensor. This system allows to determine the tridimensional movement completed by the specific camera and, estimate with a relative precision, the position of the reference points used in the process to locate itself. Furthermore, an algorithm was developed aiming for the detection of flat areas in the real world, using the previously obtained reference points, which can be considered feasible landing areas for aerospace vehicles.

All the results showed in this thesis concern to simulations obtained in controlled environments, noting that the system was not implemented in real time.

Keywords: ***Localization, Mapping, Camera, Landing Areas***

I. Introduction

Localization systems have been developed in recent years in order to improve vehicle navigation on land, air and water. However, these systems have serious limitations. The surrounding environment should be well mapped, the reference points should be well defined and all the environment physical restrictions should be well known, otherwise navigation is not possible. Because knowledge about an unknown environment is often scarce or even non-existent, navigation in that scenario implies a knowledge of the system's position in every instant relative to that same environment. On the other hand, the system's position is only correctly determined if it has mapped the environment at the same time. This localization and mapping problem was firstly studied by Randall Smith and Peter Cheeseman, in 1984, and is commonly known as SLAM[1].

This method allows estimating with considerable precision the position of a vehicle at every instant, in unknown environments, while at the same time maps that environment. Usually applied to autonomous vehicles and having the Kalman filter as the underlying technique, the SLAM method uses measurements made in the surrounding environment and reference points related with sensors' position to estimate the localization of the system where it is implemented. However, initially there is great uncertainty associated to each reference point, as the surrounding environment is completely unknown. This uncertainty will only be reduced as long as the system evolves in time, which allows determining with high enough precision in every instant its localization relative to the environment that is being mapped. This mapping is done sequentially, because the system is dependent on the sensors, which do not allow a global knowledge of the entire surrounding environment as they have intrinsic limitations.

In SLAM systems, several sensors are used. Particularly, laser sensors easily detect walls and edges and provide accurate $2D$ mapping, and sonar sensors, usually used in underwater environments, allow a precise mapping of the bottom of the sea. However, camera sensors have recently become more popular. Their light weight, low cost and real time information enable these sensors to be used in any kind of system. However, image reference point recognition as well as obtaining measurements are only possible thanks to processing techniques external to the sensor, which comes as a disadvantage. Balancing the pros and cons, using a camera allows obtaining a greater quantity of information about the surrounding environment than any other sensor as it provides a better and more realistic perception of the world. The mapping of a surface obtained with a system using a camera enables that system to have more information about the surface under study than the mapping obtained using exclusively a laser sensor or a sonar sensor.

The use of a SLAM system in aerospace applications could solve various navigation problems, specially some of the problems associated to UAV[2] navigation. Through a SLAM system it is possible for an UAV to have access to information about its position and reference points localizations at every instant. When this information is accurate and reliable enough, it could be particularly useful in UAV landing in unknown envi-

[1] *Simultaneous Localization and Mapping*

[2] *Unmanned Aerospace Vehicles*

ronments. The ability to autonomously detect suitable landing zones allows the UAV to land without human intervention in any kind of environment.
The work here presented shows the implementation of a system that estimates with precision the tridimensional movement of a camera at every time instant, maps all of the surrounding environment and detects planar surfaces which might be appropriate landing zones, resorting only to images captured by the camera. The development of this project aims at implementing this system in a quadrirotor. It enables this quadrirotor to detect a suitable landing zone and manouver during the landing process without human intervention. However, the system here presented aims only at detecting possible landing zones and no control law is proposed that would enable the quadrirotor to land using only information provided by the implemented system.
This article is divided in 4 chapters. In chapter II a description of the implementation of the method that allows for the determination of the camera's movement and the localization of the system reference points, or characteristic points, at every time instant, is made. In chapter III it is shown the algorithm that enables the detection of planar surfaces based on the system's reference points which are considered suitable landing zones. Furthermore, in chapter IV the results of the tests made to evaluate both algorithms are presented and discussed. Finally, the article is concluded in chapter V, where conclusions are taken and further work on this subject is proposed.

II. Simultaneous Localization and Mapping

The process of simultaneous localization and mapping has been developed in the last years for application in autonomous systems that require no *à priori* information in order to navigate in unknown environments. The basis of this method is the Kalman filter, which obtains estimations of the location of the system at every instant, as well as the localization of the reference points used by the filter to locate itself. A good response by the filter depends on the level of linearity of the system, being better when the system is more linear, and when the inherent noise of the system can be assumed to be zero mean white Gaussian noise. However most real systems are nonlinear and the noise associated to them can't be modulated in such a way, being necessary the implementation of another filter that allows better responses. The EKF[3] allows the estimations to be improved regarding nonlinear systems in comparison to the Kalman filter, through linearization of the functions that provide non-linearities, and basing on approximations made over the real model of the system, and despite the fact that the estimations aren't optimal.
The implemented system didn't receive any type of external input, because it didn't have incorporated any sensors that would allow the obtainment of information regarding other forces applied to the camera at every instant. Thus, the global dynamics of the system, constituted solely by the camera, can be represented from the equations that describe the process of transaction from the state $x\left(k+1\right)$ of the system and the process of the measures $z\left(k\right)$:

$$x\left(k+1\right)=f\left(x\left(k\right),w\left(k\right)\right) \tag{1}$$

$$z\left(k\right)=h\left(x\left(k\right),v\left(k\right)\right) \tag{2}$$

The function f present on the equation 1 corresponds to the function that represents the transition from the state of the system to the following state, based solely on the state of the system in the previous instant $x\left(k\right)$ and taking into account the noise associated to the process of transition, $w\left(k\right)$, that is assumed to be white Gaussian, with zero mean and an associated covariance matrix given by $Q\left(k\right)$. On the other hand, the function h, present on the equation 2, corresponds to the measuring function of the system, that allows the obtainment of all measurements on the following instant, necessary to the improvement of the estimation of the system and the reduction of the estimating error, and that is based on the state of the system $x\left(k\right)$ on that instant, taking into account the noise present on the process of measurement $v\left(k\right)$, also assumed to be zero mean white Gaussian noise, with a covariance matrix given by $R\left(k\right)$.
After the initialization of the system, this filter evolves cyclically, characterized by two phases. A first phase of prediction, where the following state is estimated, knowing only the estimation of the previous state, without any kind of measuring and a second phase of updating where the measures already made are used, in order to decrease the uncertainty and update the value of the final estimation. Thus, this filter tends to minimize the error of estimation of the system's state x(k|k) in each instant, which corresponds to the difference between the real state on an instant k and the prediction obtained from the filter to the state of the system in that same instant, and that is given by:

$$\tilde{x}\left(k\mid k\right)=x\left(k\right)-\hat{x}\left(k\mid k\right) \tag{3}$$

A. State Vector and Covariance Matrix of the System

The state vector of the system $\hat{x}$ contains the estimations of the system's state at every instant, namely the estimation of the location and orientation of the camera and of each characteristic point obtained. On the other hand, the covariance matrix P contains the values of the uncertainties that correspond to the estimations made. The state vector, as well as the covariance matrix are updated at each instant, and are defined by:

[3]Extended Kalman Filter

$$\hat{x} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_i \end{pmatrix}, P = \begin{bmatrix} P_{x_v x_v} & P_{x_v y_1} & \cdots & P_{x_v y_i} \\ P_{y_1 x_v} & P_{y_1 y_1} & \cdots & P_{y_1 y_i} \\ \vdots & \vdots & \ddots & \vdots \\ P_{y_i x_v} & P_{y_i y_1} & \cdots & P_{y_i y_i} \end{bmatrix} \tag{4}$$

where $\hat{x}_v$ corresponds to the state vector of the camera and $\hat{y}_i$ corresponds to the state vector of a characteristic point i.

The state vector of the camera is composed by 4 main vectors which contain information regarding the $3D$ position of the camera on the main frame of reference, r^W, the spatial orientation of the camera provided by a quaternion, q^{WC}, the linear velocity of the camera on the main frame of reference, v^W and the angular velocity of the camera in its own frame of reference, w^C. On the other hand the state vector of a characteristic point i is made up by 6 parameters, where $x^W_{c_i}$, $y^W_{c_i}$ and $z^W_{c_i}$ corresponds to the coordinates of the optical center of the camera on the main frame of reference in the instant when the characteristic point was observed, θ_i and ϕ_i corresponds to the azimuth and elevation angles that describe the spatial orientation of the radius m, along which the point is located and ρ_i corresponds to the inverse of the depth of the point. Both vectors can be found on the equation 5:

$$\hat{x}_v = \begin{pmatrix} r^W \\ q^{WC} \\ v^W \\ w^C \end{pmatrix}, \hat{y}_i = \begin{pmatrix} x^W_{c_i} \\ y^W_{c_i} \\ z^W_{c_i} \\ \theta_i \\ \phi_i \\ \rho_i \end{pmatrix} \tag{5}$$

This parameterization of the points made from the inverse of its depth was proposed by Andrew Davinson, on [1] and is called Inverse Depth Parameterization. Its application brings great advantages regarding the implementation of the extended kalman filter, because it allows a better estimation of the depth of point located at a great distance from the camera, where it can be represented not by a large value d, but by a small value of $\rho = \frac{1}{d}$.

B. Initialization of the System

The state vector of the camera $\hat{x}_v$ is initialized considering that the position of the camera in the main frame of reference corresponds to its origin and that the camera is not moving, being that its frame is well aligned with the main reference axis. On the other hand, the covariance matrix of the camera $P_{x_v x_v}$ is initialized considering that there's no uncertainty associated to its initial position and orientation.

The detection of a characteristic point i on the image is done through the method proposed by Harris, in [2] and it implies updating the state vector and the covariance matrix of the system. From the coordinates of the point on the image it is possible to determine the parameters θ_i and ϕ_i of the point, being that the value of ρ_i initially considered is 0.1, what makes its initial depth of $10m$. On the other hand, the updating of the global covariance matrix of the system due to the detection of the point i is made in the following way:

$$P = \begin{bmatrix} P_{x_v x_v} & P_{x_v x_v} \frac{\partial y_i}{\partial x_v}^T \\ \frac{\partial y_i}{\partial x_v} P_{x_v x_v} & \frac{\partial y_i}{\partial x_v} P_{x_v x_v} \frac{\partial y_i}{\partial x_v}^T + \frac{\partial y_i}{\partial h^i_d} R_i \frac{\partial y_i}{\partial h^i_d}^T \end{bmatrix} \tag{6}$$

where $\frac{\partial y_i}{\partial x_v}$ corresponds to the variation of the state vector y_i in relation to each of the parameters of the state vector of the camera xv, $\frac{\partial y_i}{\partial h^i_d}$ corresponds to the variation of that same vector in relation to the coordinates of the point on the plane of the image h^i_d, with distortion, and R_i corresponds to the covariance matrix associated to the noise present in the measuring process.

C. Prediction

The equation 1 represents the process of transition of system's state. Basing on the estimation of the system on the present instant, the application of the function f enables the prediction for the system's state on the next instant. Since the system doesn't get any type of input, as has already been said before, the prediction of the parameters of its state's vector is determined from a noise vector n, that contains the values for the linear and angular velocities of the system estimated on the previous instant and present on the state's vector of the system, assuming that for a small interval of time Δt, equal to $\frac{1}{30}$ seconds, these velocity values don't change significantly. This vector is regarded as noise because there is no certainty on the assumed values, and it may or may not correspond to the real velocity of the system on the following instant. The noise vector n is constituted by:

$$n = \begin{pmatrix} \hat{v}^W(k \mid k) \\ \hat{w}^C(k \mid k) \end{pmatrix} = \begin{pmatrix} \hat{v}^W_x(k \mid k) \\ \hat{v}^W_y(k \mid k) \\ \hat{v}^W_z(k \mid k) \\ \hat{w}^C_x(k \mid k) \\ \hat{w}^C_y(k \mid k) \\ \hat{w}^C_z(k \mid k) \end{pmatrix} \tag{7}$$

This way, a generic prediction of the system's state x for the next instant is given by the state's transition function f and the noise vector n:

$$\hat{x}(k+1 \mid k) = f(\hat{x}(k \mid k), n(k+1)) \tag{8}$$

The process of prediction of the state of the camera and of each one of the system's points are two different processes, hence the transition function of the system can be separated into two functions, which correspond to each of the transition processes. Applying the state

transition function of the camera f_v it is possible to obtain the predictions for the parameters that define the state vector of the camera $\hat{x}_v$. Taking into account the state's vector, containing its estimated values in a given instant k, and the noise vector n, a function f_v, linearized around the estimated values $\hat{x}_v$, is defined as:

$$\begin{aligned} f_v &= \begin{pmatrix} \hat{r}^W(k+1 \mid k) \\ \hat{q}^{WC}(k+1 \mid k) \\ \hat{v}^W(k+1 \mid k) \\ \hat{w}^C(k+1 \mid k) \end{pmatrix} \\ &= \begin{pmatrix} \hat{r}^W(k \mid k) + \hat{v}^W(k \mid k)\Delta t \\ \hat{q}^{WC}(k \mid k) \times \hat{q}'^{WC}(k \mid k) \\ \hat{v}^W(k \mid k) \\ \hat{w}^C(k \mid k) \end{pmatrix} \end{aligned} \quad (9)$$

On the other hand, the prediction of the state's vector for each characteristic point y_i consists in assuming that the point is kept at the same position in the next instant, because in reality it corresponds to static points in the world, in a way that the transition state function for each point, f_y, is given by:

$$f_y = \begin{pmatrix} \hat{y}_1(k+1 \mid k) \\ \vdots \\ \hat{y}_i(k+1 \mid k) \end{pmatrix} = \begin{pmatrix} \hat{y}_1(k \mid k) \\ \vdots \\ \hat{y}_i(k \mid k) \end{pmatrix} \quad (10)$$

Thus the prediction of the state's vector of the system $\hat{x}(k+1 \mid k)$ is completely defined, and the transition state function of the global system f, linearized, is given by:

$$f = \begin{pmatrix} f_v \\ f_y \end{pmatrix} \quad (11)$$

Moreover, the predition of the global covariance matrix of the system is given by:

$$P(k+1 \mid k) = \left(\frac{\partial f}{\partial x}\right)_{|_{\hat{x}(k|k)}} P(k) \left(\frac{\partial f}{\partial x}\right)^T_{|_{\hat{x}(k|k)}} + Q(k) \quad (12)$$

where $\left(\frac{\partial f}{\partial x}\right)_{|_{\hat{x}(k|k)}}$ corresponds to the jacobian of the transition of state of the system, linearized around the estimation of the state in the system of the considered instant and $Q(k)$ corresponds to the covariance matrix associated to the noise in the process of state transition, or the covariance matrix of f, null to all entries that correspond to the process of transition of the points.

D. Measurement's Prediction

From the prediction of the state of the camera and the prediction of the state of each point at the next instant, it is possible to obtain a prediction of the measurement that will be made by the system to every point. The prediction of the measurement $\hat{z}$ over a characteristic point i is given by the measuring function h:

$$\hat{z}_i = h_i(\hat{x}_v, \hat{y}_i) = \begin{pmatrix} \hat{u}_{d_i} & \hat{v}_{d_i} \end{pmatrix} \quad (13)$$

where $(\hat{u}_{d_i}, \hat{v}_{d_i})$ corresponds to the coordinates that are expected to be obtained from the point, with distortion, in the plane of the image. These coordinates are obtained from the transformation of the points defined on the main frame of reference ^{W} to the frame of reference of the camera ^{C} and from this to the plane of the image. The global transformation is achieved by knowing the rotation matrix of the camera R^{WC}, obtained from the components of the quaternion, that describes its spatial orientation and the calibration matrix of the camera C, given by:

$$C = \begin{bmatrix} -f_1 k_u & 0 & u_0 \\ 0 & -f_2 k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

That describes its model, containing all the parameters intrinsic to the model, and obtained from the calibration program implemented by Jean-Yves Bouguet, presented on [3]. The application of the rotation matrix R^{WC} allows the transformation of the points from the main frame of reference to the camera's frame of reference, while the application of the calibration matrix C allows the transformation of the points to the plane of the image. This process of measurement's prediction is analogous for every single point of the system. There are always values of uncertainty associated to every prediction of measurement, represented in the covariance matrix associated to the measurement's prediction, $S_{pred}(k+1)$. This matrix, that later in the phase of matching allows determining the areas on the image where it is most likely that every point is located on the next instant, is given by:

$$\begin{aligned} S_{pred}(k+1) &= \left(\frac{\partial h}{\partial x}\right)_{|_{\hat{x}(k|k)pred}} \times P(k+1 \mid k) \times \\ &\times \left(\frac{\partial h}{\partial x}\right)^T_{|_{\hat{x}(k|k)pred}} + R \end{aligned} \quad (15)$$

where $\left(\frac{\partial h}{\partial x}\right)_{|_{\hat{x}(k|k)pred}}$ corresponds to the variation of the measuring function h, linearized around the value of the prediction of state of the system to the next instant and R corresponds once again to the covariance matrix associated to the error in the measurements. For each point I considered, the matrix defined in 15 corresponds to a square matrix S_{pred_i}, with dimension 2×2, whose diagonal contains the values of the variances associated with $\hat{u}_{d_i}$ and $\hat{v}_{d_i}$, from the equation 13.

E. Matching

From the variance values given by the diagonal of the matrix S_{pred_i}, it is possible to generate a trust zone, with dimension $n \times n$, around the values of the prediction where there's a probability of approximately 95% of the point i being found. Taking into account

that every characteristic point i is characterized by a 21×21 pixels window, named $patch_{i_{real}}$, as can be seen on figure 1 and applying a crossed correlation between each $patch_{i_{real}}$ and every n^2 windows with the same dimensions obtained from that trust zone, called $patch_{i_{pred}}$, it is possible to obtain the real value of those measurements, z, for each point, considering that the correlation will be as big as the similarity between each $patch_{i_{real}}$ and $patch_{i_{pred}}$ considered, reaching a maximum value in the case of total correspondence between them.

However, in case of a point i isn't found inside this zone, it is considered that there is no correspondence between the prediction and the real measuring for that same point, being that the components of S_{pred_i} are eliminated from the covariance matrix $S_{pred}(k+1)$. This way, the matrix $S_{pred}(k+1)$ will be only a container of the information that corresponds to the points that had a correspondence, thus representing the covariance associated with the difference between real measurements and the predictions of those measurements, in other words, the error associated to the measuring. This difference between real measurements and its prediction corresponds to the innovation process of the system and it's covariance matrix is thus defined by $S(k+1)$.

F. Update of the System State

The last step on the implementation of the extended kalman filter corresponds to the update of the state's vector of the system $\hat{x}$ and its covariance matrix P, from the knowing of the real measurements of the system and the prediction of its state for the next instant. Its update is thus given by the equations 16 and 17:

$$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + K(k+1) \times \times \nu(k+1) \quad (16)$$

$$P(k+1 \mid k+1) = P(k+1 \mid k) - K(k+1) \times \times S(k+1) K^T(k+1) \quad (17)$$

where the innovation process $\nu(k+1)$ is given by:

$$\nu(k+1) = z(k+1) - \hat{z}(k+1) \quad (18)$$

and where $K(k+1)$ corresponds to the optimal Kalman gain, that ensures the minimization of the estimation error, given by:

$$K(k+1) = P(k+1 \mid k) \times \left(\frac{\partial h}{\partial x}\right)^T_{|_{\hat{x}(k|k)match}} \times \times S(k+1)^{-1} \quad (19)$$

G. Scale Factor

The method that has been presented and described on the previous sections allows to provide information regarding the displacement done by the camera and the spatial location of every points used as a reference. However, this method doesn't have the ability to understand if the movement is a real displacement, or if the arrangement of the points is in fact its real arrangement. Despite the fact that the relation between every measurement is correct, on such a system it is always necessary to define a scale factor that allows the scaling of every measurement done by the system. The scaling of these estimations can be done initializing the system with specific characteristic points, whose coordinates are known, with a null value of uncertainty, scaling all measurements from the known dimensions of an object that appear on the image, after all estimations have been made or through the utilization of auxiliary sensors, as altimeters or accelerometers, that enables a greater certainty in some of the measurements made. Thus, every other measurements and estimations done by the system will be scaled and will correspond to their real value.

III. Algorithm for Landing Zones Detection

The system described in section II estimates accurately the camera's movement over time, regardless of the trajectory performed by the camera. This kind of system can be implemented in a UAV in order to get information on the position, orientation and linear/angular velocities. Moreover, this system is able to map the surrounding environment with great precision, and it is this information that will be available to enable the UAV to detect autonomously plane surfaces that are eligible as landing zones.

Regarding the implementation of this system in a quadrirotor, the camera would have to be placed on the lower part of the vehicle with the Z axis oriented towards the floor and the X and Y axis parallel to this, so that the camera is able to determine its distance to the floor. Thus, the main reference frame will be centered in the camera's optical center and consequently, in the quadrirotor, and all the estimated points would be represented in this reference frame.

The landing zone detection algorithm here implemented is based in the RANSAC[4] algorithm, described in [4], which is also used to determine planes in space. This algorithm uses a collection of reference points obtained by the system to detect plane surfaces in the world, where each group of three points in space define a plane. For the spherical coordinate representation of each detected plane, in terms of ρ, θ and φ, the following equation holds:

$$\rho = (\cos\theta\cos\varphi) X + (\sin\theta\cos\varphi) Y + (\sin\varphi) Z \quad (20)$$

[4] Random Sample Consensus

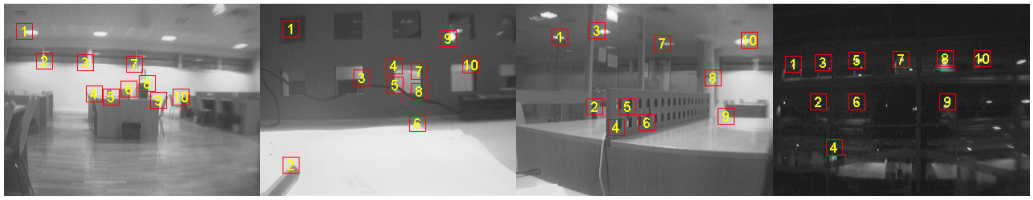

Figura 1: Characteristic points detected in different scenarios, being that in each case were detected 10 points with the desired characteristics

Analyzing the distance between every point and each of the planes, it is possible to see which points belong to a certain plane if its distance to the plane is lower than a predetermined limit value. On the other hand, the considered planes correspond only to those which contain the most number of points. The minimum of points necessary for a plane to be considered is also predetermined. All these restrictions make the algorithm more efficient because only the best planes analyzed are considered.

However, between the planes considered it is necessary to pick the one which can be a good landing spot. A good landing spot will be a plane defined by an angle φ approximately equal to 90^o. An alternative parameterization where the X and Z axis are interchanged may also be used: in this case the good plane is selected if θ and φ are approximately equal do 0^o. Because this algorithm is based in the points obtained by the system to determine the best plane which suits these points, the error associated to the determination of the plane depends on the error of the point position estimation. Therefore, it is necessary for the points used to be reliable and with little uncertainty associated, as landing spot determination requires the maximum precision possible.

IV. Experimental Results

Two experimental tests were conducted aiming to evaluate the track and mapping method discussed in chapterII, as was the landing area detection algorithm present in chapter III. All tests were taken in controlled environments, in a sequential way, allowing the complete and non-erroneous implementation of the previously presented algorithms.

A. Experimental Test I

In the first test, the camera was initially set at $1m$ from the two panel intersection, placed both perpendicular and vertically at one another, as shown in figure $3a)$. The camera's movement consisted in it's displacement in the X axis positive direction, approximately along $0.50m$, followed by a return to the original position and a movement's continuation this time in the negative axis direction, stopping ate $0.10m$ from the axis origin. Noting the points distribution detected throughout the panels, present in figure $3b)$, and the displacement suffered by the camera, one realises that the biggest parallax angle obtained was approximately of 26^o for the points located near the panel's intersection. This angle tends to become smaller as the points grow further away from the image centre. This value corresponds to almost half of the desired optimum value for a parallax angle, of 45^o, needed for an uncertainty reduction associated with the depth's estimation for minimum values. It is always necessary that the maximum displacement of the camera is the same order of magnitude of the maximum depth of points, otherwise, the parallax angle will constantly be small and the achieved estimations will always bear an uncertainty value. The movement estimation in the X axis, pictured in figure $3c)$, although being quite accurate, contained associated errors that could not be ignored, as the maximum value for the standard deviation was approximately of $0.044m$, which corresponded to the camera's movement inversion. Likewise, the depth's estimation of each system's point was affected by an uncertainty value rather large as one can see in the covariance ellipsis of figure $3d)$, being reduced for points 15, 16, 17 and 18, initially centered in the image, which were the ones that felt the most the camera's movement. As previously mentioned, these estimations could be improved, if the displacement was in fact bigger.

An effort was made, throughout the test, for the camera don't suffer any kind of movement in the Y and Z axis. Such care was taken in order for the global movement through the x axis could be considered as linear as possible. In spite of this, as the test was conducted by a human hand, some movement in the non-desired axis occurred, although it presented small values, as shown in figures $3a)$ and $3e)$, being the maximum variation in the Y axis of approximately $0.06m$ and of $0.11m$ for Z axis, both around the optimum value, $0m$.

Similarly, the disposition of the points on the vertical plane XY is quite close of their real disposition, as it can be observed in figure $3f)$, being confirmed the ground proximity of points 4, 8, 13 and 16. Noting that the estimated values that describe the camera's movement and the location of each point in space are scaled, having obtained the scale factor considering the known dimensions of each black rectangle, present in both panel's pattern. Although the uncertainty in the points depth determination was not reduced to low optimal values, it was applied the flat zone determination algorithm to the set of points gathered in this

test. Bearing in mind the alternative parameterization present in chapter III, with X and Z axis inversion, the plane containing the major number of points presented a θ value of approximately 45 ± 5^o and a φ value of roughly $0^o \pm 5^o$, relating to the camera's initial instant. On the other hand, the plane obtained using the algorithm presented θ and φ values of approximately 11.9775^o and 21.2374^o, respectively, values that distance a bit from the real values. As it can be observed in figure $3g)$ and $3h)$, there is some inclination present in the plane but the associated error is still too high comparing with the real values, being around 33.0225^o for θ and 21.2374^o for φ. It is imperious to realize that in the initial instant, the camera's spatial orientation might have had some associated error, ist est., initially the camera might not have been observing the plane with a θ angle of 45^o, but suffering from an unknown inclination, that could be in fact bigger or smaller than the ideally considered. This, in combination with low parallax angles obtained by an insufficient camera movement, allowed the formation of a plane whose parameters varied a bit from the expected values, but presented itself as the best plane from the analyzed set point, as can be verified by figures comparison $3g)$ and $3h)$.

B. Experimental Test II

In the second experimental test, the camera was initially at $1.15m$ from the panel, placed both vertical and perpendicularly to the camera, as shown in figure $3a)$. The movement made by the camera consisted in moving it again in the X axis positive direction, approximately by $0.90m$, followed by a return to the initial position and then continue the dislocation in the same axis, this time in the opposite direction, by $0.40m$, after passing through the origin point. Contrary to the first test, in this case the displacement suffered by the camera was enough to produce high parallax angles, being the maximum value of approximately 38^o, nearby the optimum value. This allowed obtaining the correct depth estimation of the points, with low uncertainty values, as desired.

The camera's movement estimation through the X axis, as pictured in figure $3c)$, is quite similar to the real movement, with small associated errors, especially when the system reaches stability, achieved when the camera switches the movement direction. From that instant forward, the maximum value for the standard deviation was around $0.0208m$, a rather low value which allows one to be quite confident about the obtained estimation. Likewise, the depth's estimation of each system points was affected by low uncertainty values, pictured in the covariance ellipsis of figure $3d)$, being the maximum value of the standard deviation of approximately $0.0141m$, an equally low value.

Once again, during the test, an effort was made for the camera not to suffer any kind of movement in the Z and Y axis, in order to the global camera displacement be as linear as possible along the X axis. However, it revealed to be impossible to avoid the camera's movement in the other axis, although through low values, as shown in figures $3a)$ and $3e)$, being the maximum variation in the Y axis of $0.1396m$ around $0m$, and through the Z axis a variation of $0.0768m$, also around the optimal value of $0m$. These values are acceptable considering the kind of movement involved. As in the first case, the point's disposition in the vertical plane XY is indeed similar to theirs real disposition, as one can see in figure $3f)$, being confirmed once again the ground's proximity of points 4, 10 and 16. It is important to realise that the estimated values that describe the camera's movement and the location of each point in space are equally scaled, having the scale factor been obtained through the method describe in the first test.

This case corresponds to the ideal case, where the real plane represents a possible landing zone, because it presents itself parallel to the image plane and the low uncertainty associated with the obtained estimations allowed to determine a flat zone similar enough to the real case, applying the implemented algorithm. Noticing again the plane's alternative parameterization, it presented a real θ e φ values of approximately $0^o \pm 5^o$, in what concerns to the initial camera's instant. In its own way, the plane obtained using the algorithm presented values for θ e φ of approximately 4.4811^o e 0.3402^o, respectively, values that are rather close to the real values. As it can be observed in figure $3g)$ and $3h)$, the obtained plane fits perfectly with the considered points, as the errors in its determination are of the same magnitude as the real measurement errors, being that way the best plane for the analysed set points and that can possibly be considered as a good landing plane.

C. Discussion of the obtained results

Through the results analysis clearly it can be considered that the results obtained for the second experimental test were better than the one's from the first, mostly due to the increased displacement made by the camera. This points out the inherent difficulty in determining the point's depth, always being necessary to have big parallax angles that can lead to estimated values that are quite approximated from those real. These tests allowed to test the track and mapping method simultaneously, where one considers the achieved results as being fairly good, whose estimations are close to the real values, especially in the second considered case. The good parameters' estimation, namely the correct location of the points in space, is the principal feature for the algorithm of flat zone detection can generate good landing areas, with great precision.

The obtaining of flat zones depends on the point's estimated value in the instant of lowest uncertainty. This

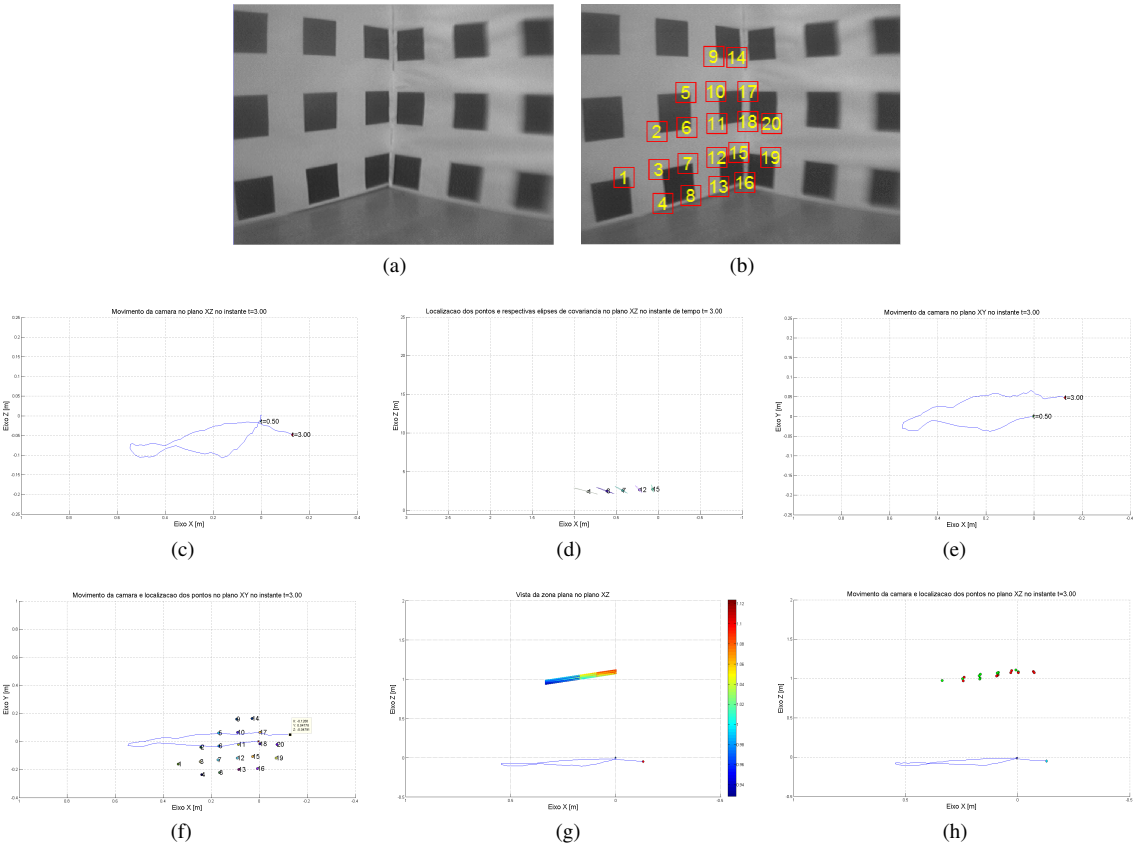

Figura 2: a) Scenary of the first test, with two panels arranged vertically, perpendicular to each other; b) Characteristic points obtained by the system; c) Global movement of the camera, along the X axis, in the horizontal plane XZ; d) Estimation of the points depth, in the final instant, with the covariance ellipses that represent the estimation uncertainty; e) Estimation of the camera's global movement , in the vertical plane XY; f) Estimation of the points location in the vertical plane XY, in the final instant; g) Obtained plane for the group of points of the system, with a viewing from the horizontal plane XZ; h) Group of points of the system that allowed the plane's determination, with a viewing from the horizontal plane XZ

way, if that uncertainty value is not low enough for the estimation to be considered optimal, the obtained planes may not correspond entirely to the real case, as happened in the first situation. On the other hand, in the second case the uncertainty value associated with the point's estimation allowed the generated plane to be relatively similar to the real plane, as in this case the plane was in an almost vertical/perpendicular position, relating to the camera.

The tests conducted also allowed to test the plane detection algorithm and it can be considered a rather successful test, with low associated errors. This algorithm is capable of detecting flat zones in the real world, which may or may not be considered landing zones, depending on the points that define those areas.

These results sustain the claim with a certain degree of safety that the implementation of these algorithms produces good results, as a system of this kind supplies a fairly amount of information to the vehicle to whom it is attached to. Once again it is stressed that there were no other sensors associated to this system, which could decrease the estimation errors if they were used.

V. Conclusions

A system which determines with relative precision the tridimensional movement of a single camera as well as the localization of the reference points used by the camera to position itself has been successfully implemented in this work. However, it was difficult to determine the depth at which the points are located based on the analysis of the images obtained by a single camera. The movement of the camera is critical because the movement has to be such that the uncertainty in the point estimation is reduced to optimal values. In some systems, stereo vision, consisting in two cameras placed at a known distance between each other and with a known angle, allows for the determination of the depth of the points within a certain vision radius, without any required system movement. This certainty in the initial depth estimation thus provides correct information that enables to determine accurately its trajectory, as the uncertainty associated with the initial points is almost zero.

Another algorithm which uses the same reference points to determine plane surfaces in the world which

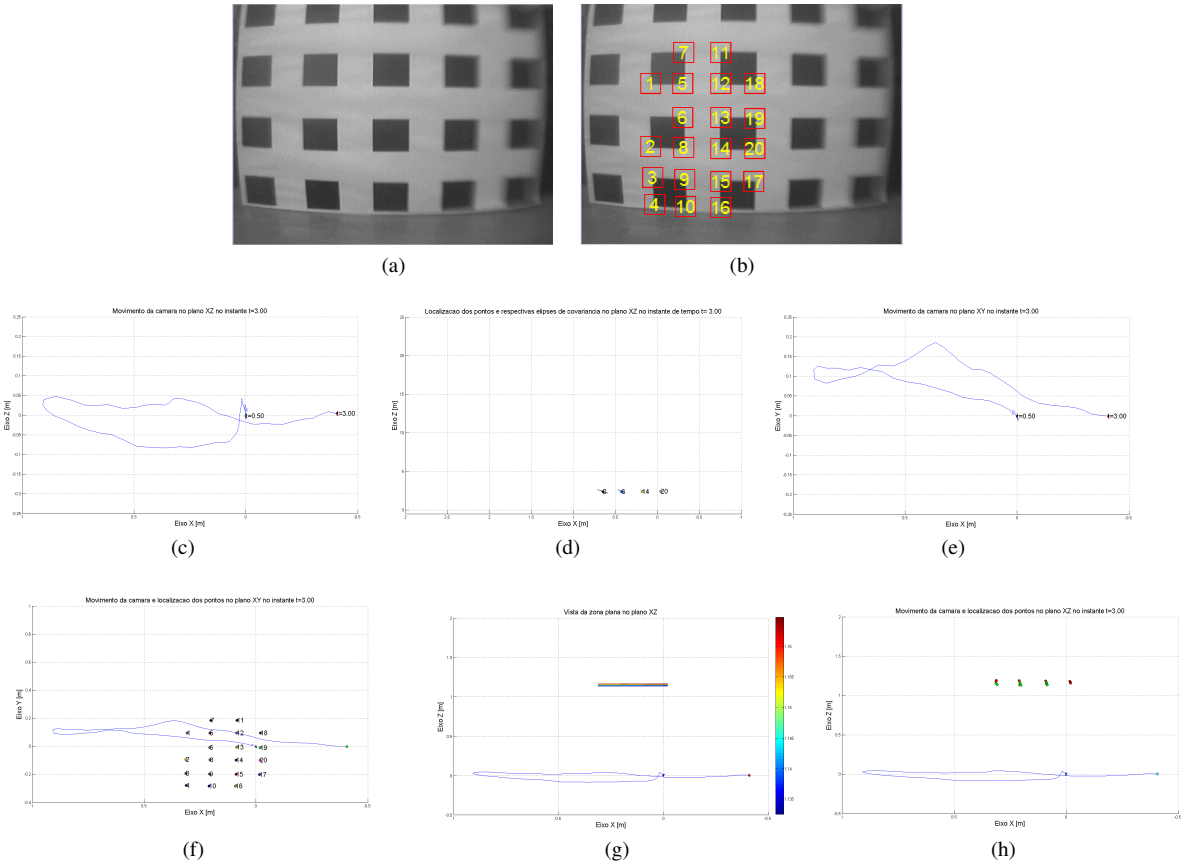

Figura 3: a) Scenary of the second test, with one panel arranged vertically, parallel to the camera; b) Characteristic points obtained by the system; c) Global movement of the camera, along the X axis, in the horizontal plane XZ; d) Estimation of the points depth, in the final instant, with the covariance ellipses that represent the estimation uncertainty; e) Estimation of the camera's global movement , in the vertical plane XY; f) Estimation of the points location in the vertical plane XY, in the final instant; g) Obtained plane for the group of points of the system, with a viewing from the horizontal plane XZ; h) Group of points of the system that allowed the plane's determination, with a viewing from the horizontal plane XZ

might be considered suitable landing zones for UAVs has also been successfully implemented. The ability of the system to detect these landing zones without human intervention enables the system to be autonomous and operate in unknown regions without incurring in the risk of needing to land and not be able to land due to lack of information on the surrounding terrain.

Despite the overall results being good, this system was tested only in controlled environments, where the movement of the camera was known and considered slow. The system was not tested in a real scenario, namely it was not applied to a quadrirotor, where the image change rate is greater. Some changes to the actual implementation would have to be done so that the system response was acceptable under those new conditions. Regarding this fact, although the work done successfully accomplished most of its goals, it is not fully finished.

A. *Future Work*

The actual system and algorithms do not need improvement as they already produce very good responses, as shown. However, there is room for improvement before this system can be applied to a quadrirotor, as real time responses are needed. The presented system determines the landing zones with precision, but this information needs also to be included in a controller, so that the quadrirotor is able to land in a suitable manner. The development of this controller, as well as the implementation of this system in real time on the quadrirotor are important follow-up steps in case a complete and real application of the system is desired.

REFERENCES

[1] J. M. M. M. Javier Civera, Andrew J. Davison, "Inverse depth parametrization for monocular slam," *IEEE TRANSACTIONS ON ROBOTICS*, vol. VOL. 24, pp. 932–945, 2008. 3

[2] M. S. Chris Harris, "A combined corner and edge detector," *The Plessey Company*, p. 147Ű152, 1988. 3

[3] J.-Y. Bouguet. (2008) Camera calibration toolbox for matlab. 4

[4] P. G. F. Tarsha-Kurdi, T. Landes, "Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data," *IAPRS*, vol. XXXVI, p. 6, 2007. 5

[5] N. M. O. S. Andrew Davison, Ian Reid, "Monoslam: Real-time single camera slam," *IEEE*, vol. 29, p. 16, 2007.

[6] D. W. M. Andrew J. Davison, "Simultaneous localization and map-building using active vision," *IEEE*, vol. Vol. 24, pp. 965–880, 2002.

[7] J. Aulinas, "3d visual slam applied to large-scale underwater scenarios," Master's thesis, Institute of Informatics and Applications, University of Girons, 2008.

[8] D. S. Darya Frolova, "Matching with invariant features," Master's thesis, The Weizmann Institute of Science, 2004.

[9] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," p. 8.

[10] ——, "Slam with a single camera," Master's thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, 2002.

[11] ——, "Mobile robot navigation using active vision," Master's thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, 1998.

[12] K. G. Derpanis, "The harris corner detector," 2004.

[13] E. B. Joaquim Salvi, Yvan Petillot, "Visual slam for 3d large-scale seabed acquisition employing underwater vehicles," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 22–26, 2008.

[14] M. G. Lucas Teixeira, Waldemar Celes, "Accelerated corner-detector algorithms," Master's thesis, Tecgraf - Computer Science Department.

[15] J. D. Luiz Mirisola, "Uma metodologia de odometria visual/inercial e slam 3d com um vant," Master's thesis, Divisão de Robótica e Visão Computacional - DRVC, CTI - Centro de Tecnologia da Informação, Brasil; Instituto de Sistemas e Robótica, Universidade de Coimbra, Portugal.

[16] S. S. Mitch Bryson, "Bearing-only slam for an airborne vehicle," Master's thesis, ARC Centre for Excellence in Autonomous Systems, Australian Centre for Field Robotics, University of Sydney.

[17] J. Neira, "Pure visual slam," p. 53, 2008.

[18] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," Master's thesis, Institute for Systems and Robotics, Instituto Superior Técnico, 2004.

[19] S. Rotenberg, "Orientation & quaternions," 2005.

[20] I. M. Simon Baker, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*.

[21] D. M. Somkiat Wangsiripitak, "Avoiding moving outliers in visual slam by tracking moving objects," *IEEE*, 2009.

[22] J. S. D. C. Zhenyu Yu, Kenzo Nonami, "3d vision based landing control of a small scale autonomous helicopter," *International Journal of Advanced Robotic Systems*, vol. Vol. 4, No. 1, p. 51Ű56, 2007.