COMPUTER VISION BASED MONTE CARLO LOCALIZATION FOR AUTONOMOUS AERIAL VEHICLES

DOUGLAS G. MACHARET*, ARMANDO A. NETO*, VÍCTOR C. DA S. CAMPOS*, MARIO F. M. CAMPOS*

* *Vision and Robotics Laboratory (VeRLab)*
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil

Emails: doug@dcc.ufmg.br, aaneto@dcc.ufmg.br, kozttah@gmail.com, mario@dcc.ufmg.br

Abstract— A robot's knowledge of its location is a fundamental information on mobile robotics, allowing greater autonomy in decision-making problems. Thus, this paper proposes a technique for localization of Unmanned Aerial Vehicles based on Computer Vision sensing and using the Monte Carlo Localization method. Only natural landmarks already present in the environment are used, avoiding the need for manual insertion of recognizable landmarks. For the identification of landmarks the Scale Invariant Feature Transform algorithm (SIFT) is used. We present results from a model of a small autonomous aerial vehicle in a non-structured environment.

Keywords— MCL, SIFT, UAV

1 Introduction

The interest and research in Unmanned Aerial Vehicles (UAVs) have been growing, specially due to the decrease in cost, weight, size and performance of sensors and processors. Clearly UAVs have their niche of applications, which cannot be occupied by other types of mobile robots, since they are capable of covering a broad set of relevant applications. They are able to navigate over large areas much faster than land vehicles, with a privileged view from above, which is one of their main uses in monitoring and surveillance. As the availability increases, so does the possibility of having multiple such vehicles traversing a given volume of the space. Therefore, there is growing need to study and develop techniques for the operation and navigation of different types of aerial robots.

Several open issues still demand much research effort in order to endow a aerial vehicle with full autonomy in a real environment. Among those, one well known problem is posed by one of mobile robotics basic questions related to navigation: "Where is the robot?" Throughout the years, several efficient and relevant solutions have been proposed with varying degrees of success.

In order to fulfill tasks such as monitoring and surveillance it is of extreme importance that the aerial vehicle is capable of localizing itself without human intervention. In most cases GPS (*Global Positioning System*) devices are used, however the quality of the information obtained through this system is affected not only by the number of satellites in sight and the quality of the received signal but it might also suffer eletromagnetic interference from other devices which could also degrade the obtained data. In addition to this, an important factor to be considered especially in the case of aerial vehicles is the amount of weight it can carry.

UAVs can be divided into at least three classes: rotary-wing aircrafts (e.g. helicopters and quadrotors), aerostatic aircrafts (such as aircrafts and hot air balloons) and fixed-wing aircrafts (airplanes). The technique described in this text will be instantiated for fixed-wing UAV, however, without loss of generality, it can be applied to other types of vehicles. Fixed-wing aircrafts present constraints on their mobility such as minimum curvature, maximum angle of climb or dive, and minimum speed.

Considering the restraints that have been raised, it led us to think in using a sensor that would be usable in more than one task, in this case a camera, that could be used for both monitoring and localizing the aircraft.

Initially, for our vision-based localization approach, a match between features in the present scene and features in a map covering the whole region where the vehicle will move is carried out. In this work, we identify the features using the Scale-Invariant Feature Transform (*SIFT*) algorithm, which will be described in Section 3.1.

The next step consists of using the data obtained from the vision system for the position calculation. Many methods can be found in the literature such as Kalman Filtering, Bayesian Methods and Particle Filters. The method chosen in this paper is named Monte Carlo Localization (*MCL*) and it will be detailed in Section 3.2.

The choice of this method over the others was due to some advantages it has (Dellaert et al., 1999), among which there are:

- Ease of Implementation;
- Unlike the Kalman Filter, it is able to represent multi-modal distributions;
- The state represented in the samples is not discretized and it demands less memory and processing (in case the number of particles is less than the number of cells) than grid based methods (such as Markov Localization).

Therefore, this paper proposes an UAV localization through the use of the *MCL* algorithm and computer vision. The rest of this paper is organized as follows: Section 2 presents an overall view related works found in the literature, next in Section 3 the localization stages are presented. Sections 4 and 5 present the experiments, the discussion, and analysis of the results obtained with the proposed method, applied to a model of a real UAV developed on the Federal University of Minas Gerais/Brazil.

2 Related Works

2.1 Ground Robots

Many works can be found in the literature that deal with the problem of localization using Computer Vision. However, a large amount of these are directed towards ground robots. Among these there are those which use only the information obtained through Computer Vision in order to localize, as seen in (Ramisa et al., 2008), where the authors use various feature region detectors to carry through a topological localization of the robot, and (Zhang and Kosecka, 2006), in which a georeferenced image database is used alongside the *SIFT* algorithm to infer the robot position from the current sight.

There are also works that make a sensor fusion between the data given by the camera and other sensors present in the robot. In (Bennewitz et al., 2006), a particle filter and a SIFT descriptors map of the environment are used to calculate the 3 degrees of freedom pose of the mobile robot. In (Leung et al., 2008), a map consisted of aerial images and a particle filter are used to estimate the 3 degrees of freedom pose of the mobile robot given ground level images and the robot odometry. Whereas in (Elinas and Little, 2005), the authors carry through a Monte Carlo Localization of the robot with 6 degrees of freedom using a stero vision system.

Moreover, there are the works in which the robot does not possess a previous map of the environment, and, therefore, deal with the problem of Simultaneous Localization and Mapping (SLAM). Among these we can cite (Barfoot, 2005) and (Zhang et al., 2006), which uses a ground robot and the *SIFT* algorithm to carry through the localization and mapping using a combination of particle filter and EKF, and (Se et al., 2001), which uses 3 cameras, the robot odometry and the *SIFT* algorithm in a method developed by them in order to construct a 3D descriptor map and localize the robot simultaneously.

2.2 Aerial Robots

Despite being relevant, all works present thus far have dealt with the use of ground robots. We will now deal with the case of aerial robots. It is possible to find works in the literature which deal with the problem of localizing unmaned aerial vehicles (UAVs), but we are especifically interested in works that proposes solutions to this problem using Computer Vision.

The first works we will approach are those in which known beacons are displaced in the scene and using the camera present in the vehicles identify and use the beacons to localize the aircraft. Among these we can cite (Bath and Paxman, 2005), (Campos and de Souza Coelho, 1999) and (Saripalli et al., 2002). However, this is not a good approach since it demands a change in the environment the vehicle will be acting, reducing the applicability of aircrafts in regions that are unattainable by land.

Another group of works, in which this paper fits, consists of using beacons already present at the environment, making it possible to use the vehicle in any environment without previous human interference. As an example of such works we can cite (Kong et al., 2006; Sinopoli et al., 2001; Bensebaa and Martins, 2008; Caballero et al., 2008; Wu and Johnson, 2008; Steffen and Forstner, 2008; Karlsson et al., 2008). These works use computer vision to calculate visual odometry or a higher level localization as in this paper.

3 Methodology

3.1 SIFT

The identification of the matching points between images is of great utility in Computer Vision problems. In this article, the main function of the vision based system is, given the aircraft vision (embedded camera), to identify feature points in the environment and match them against feature points in a map. In order to solve this problem, the descriptors obtained through the *SIFT* (*Scale Invariant Feature Transform*) (Lowe, 1999) algorithm will be used.

The *SIFT* descriptors of a given image can be understood as features of this image. The choice of this algorithm was led by the fact that these descriptors are invariant to changes in scale, rotation, translation and luminosity, making them robust to the movements of an aircraft during a flight. The algorithm is composed of four stages:

1. Scale-space extrema detection: In this stage, a search for local maxima and minima is carried out on the Difference of Gaussian (DoG) images at multiple scales. DoG images are obtained by taking the difference of successive Gaussian-blurred images.

2. Keypoint localization: For each extrema location detected, a detailed model is ajusted in order to determine the exact location and the scale. Keypoints are selected based on stability measures. In this stage the best points for the mapping system are defined, by means of gradient measures.

3. Orientation assignment: Each keypoint's orientation is assigned by means of the image local gradients. From there on, every operation will be made in relation to image data transformed according to each keypoint's orientation and scale. By doing so, invariance to these transformations are obtained.

4. Keypoint descriptor: Each keypoint local gradient is measured, by using its neighborhood. These measures are then transformed to a representation that allows tolerance to significant levels of distortion and light changes.

In the following section we present how these descriptors are going to be used in the *MCL* particle weighting.

3.2 Monte Carlo Localization

In order to start solving the problem, we consider that the vehicle possesses a map of where it will navigate and also that it can be found somewhere within the map. Since we are dealing with an aircraft there is no need to check wether a region is free or not, considering that the airspace is always free.

The *MCL* method represents the vehicle position on the environment by means of a fixed number of particles, and each of these represents a possible aircraft position and have a weight associated representing the belief that the aricraft is really in that position. Each particle will have its position represented by $\begin{bmatrix} x & y \end{bmatrix}$, disconsidering its orientation.

The method can basically be divided in three distinct stages that iterate during the navigation, they are:

- Prediction: Based on a transition model, predicts the robot position after a certain action;
- Update: According to the observation model, weights (beliefs) are calculated for each particle present on the environment;
- Resampling: The particles are weighted resampled based on the weights calculated on the previous stage.

The *MCL* algorithm used on this paper was taken from (Thrun et al., 2005) and is presented in a more formal way on Algorithm 1.

The algorithm execution is as follows, initially, a given M quantity of particles is randomly distributed over the map, and all of them have $\frac{1}{M}$ as the weight value.

Next, during the prediction stage, all particles receive the control parameter that was used on the aircraft locomotion, known in this paper as $u_t = \begin{bmatrix} v & \theta & \psi \end{bmatrix}$, where v represents the aircraft's ground speed,θ the pitch angle and ψ the yaw angle. Each particle's new position is then calculated using the transition model by Equations 1a and 1b.

Algorithm 1 MCL($\mathcal{X}_{t-1}$, u_t, z_t, m)

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1$ to M **do**
  $x_t^{[m]} = \mathbf{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$
  $w_t^{[m]} = \mathbf{measurement\_model}(z_t, x_t^{[m]}, m)$
  $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \left\langle x_t^{[m]}, w_t^{[m]} \right\rangle$
end for
for $m = 1$ to M **do**
  draw i with probability $\propto w_t^{[i]}$
  add $x_t^{[i]}$ to $\mathcal{X}_t$
end for
return $\mathcal{X}_t$

$$x_t - x_{t-1} = v\, cos(\psi) cos(\theta)\, \Delta t + \delta x \tag{1a}$$

$$y_t - y_{t-1} = v\, sin(\psi) cos(\theta)\, \Delta t + \delta y \tag{1b}$$

In the calculation of the new position we add a δx and a δy factor as a way to simulate the uncertainty of the sensor readings. In case that the new position of a particle is outside the map, a new random position is sampled within the map boundaries.

Next, we apply the observation model, responsible for calculating the weight of each particle. For each particle in the map, the aircraft's field of view in that position is recovered and its *SIFT* features are matched against the features on the aircraft's real view. The weight is then given proportionally to the number of matching points in relation to the number of descriptors found on the aircraft's view.

In case no descriptor is calculated on the aircraft's view or there are no matches, the particle receives a really small weight value.

The particles weighted resampling is then carried out through Algorithm 2.

Algorithm 2 low_variance_sampler($\mathcal{X}_t$, $\mathcal{W}_t$)

$\bar{\mathcal{X}}_t = \emptyset$
$r =$rand$(0; M^{-1})$
$c = w_t^{[1]}$
$i = 1$
for $m = 1$ to M **do**
  $u = r + (m-1) * M^{-1}$
  while $u > c$ **do**
    $i = i + 1$
    $c = c + w_t^{[i]}$
  end while
  add $x_t^{[i]}$ to $\bar{\mathcal{X}}_t$
end for
return $\bar{\mathcal{X}}_t$

4 Experiments

Our technique was used to locate a simulated small unmanned aerial vehicle. That robot was modeled as a fixed-wings aircraft based on an UAV named AqVS (Figure 1), developed at Universidade Federal de Minas Gerais/Brazil. This is a small hand launched hybrid electric motor sail plane, equipped with barometric altimeter, infrared inclinometer, airspeed sensor and CCD camera, and controlled by a set of PID stabilizators running on a Palm® PDA for autonomous navigation (Iscold, 2007). The AqVS presents the following characteristics:

- curvature minimum radius: about 15 meters,
- maximum cruising speed: approximately 50 km/h,
- uncertainty of localization: 12 meters.

The above values were determined using data from actual flights, considering a speed of approximately 50 km/h. This vehicle has shown to be a good choice for testing our methodology because of its large uncertainty of localization.

Figure 1: AqVS, an UAV from the Universidade Federal de Minas Gerais-Brazil.

At first, satellite images (GMA, 2008) of a certain region were collected on the highest resolution available so as to represent the place where the aircraft would navigate. After gathering all of the images, they were united in a mosaic, forming an image that covers a big area while also maintaining the details.

The aircraft mathematical model was implemented in Matlab®, and the flight path was instantiated by a virtual camera inside a 3D modelling software. Synthetic images were captured from this camera simulating the view of a real aircraft over the monitored region. The previously built map was established as a stage for the flight.

The results obtained through this framework were satisfactory, allowing the execution of initial experiments of algorithms that are to be embedded in aerial vehicles without the need of the real equipment. Figure 2 presents a simulated view of the aircraft camera.

In order to validate the proposed method the following experiment was perfomed, the aircraft should move by a trial known trajectory (trial since there is noise aggregated to the aircraft localization information), checking wether it is possible to obtain a good precision in the calculation of the aircraft position by using the proposed method. Figure 3 presents the calculated trajectory over the map exhibiting the regions where the aircraft is meant to pass.

Figure 2: Simulated aircraft camera view

Figure 3: Vehicle's estimated trajectory.

It is important to note that, during the experiment, the aircraft also has a variation in its altitude. That is done so as to make it possible to analyse the scale variation of the images collected by the camera. The aircraft's altitude over time is shown in Figure 4, the measurement noise is also seen in it.

Differently from other approaches found on the literature, there is no need for the particles to occupy pratically every available space on the map, that is due to the fact that each particle possess a field of view (in this paper represented as a 100x100 window), and, in case many particles are close, unnecessary calculations will be carried out without many contribuition to the localization calculation. Therefore, in the experiment described here, 300 particles were distributed over a 1127x1236 map.

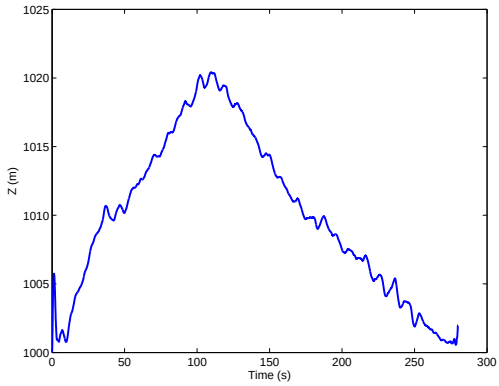

Figure 4: Altitude variation during the experiment.

Figure 5 presents the particle's behavior over time during the experiment execution. It is possible to note that initially all of the particles are scathered randomly over the map 5(a). After a determined period of time it is possible to note the disappearance of particles over certain regions 5(b), as is the case of the particles that were above a lagoon (water), this can be mainly explained by the fact that the water surface is mostly uniform, which mackes the *SIFT* find few descriptors and these particles recieve a small weight since the aircraft is flying over a region with many identifiable descriptors.

The particles gather over the higher weight points until they form a point cloud. This point cloud keep on locomoting according to the transition model. The particles do not gather at only one point mainly due to modeled sensors errors.

5 Conclusions and Future Works

The results obtained from the proposed method were fairly gratifying, allowing us to estimate the global position of the aircraft in a given map region, without previous knowledge of its starting position.

However, the calculation of the matching points spent a fair amount of computing time, not being as efficient as desired. Therefore, as a next step it is necessary to optimize the method in a way that it runs in minimun time without losing quality. Still related to the computaional cost involved, it would be interesting to carry through a better analysis of the number of particles and vision window size relation so that each iteration runs faster while also assuring the convergence time of the particles in a certain region.

In addition to this there is the intention of adding the orientation angles and the height variantions (Z axis) to the aircraft's state vector.

We also hope to carry through the experiments with real aircrafts, allowing, therefore, a better analysis of the method's behaviour amidst the real world difficulties.

References

Barfoot, T. (2005). Online visual motion estimation using fastslam with sift features, *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on* pp. 579–585.

Bath, W. and Paxman, J. (2005). Uav localisation & control through computer vision, *Proceedings of the Australasian Conference on Robotics and Automation*.

Bennewitz, M., Stachniss, C., Burgard, W. and Behnke, S. (2006). Metric localization with scale-invariant visual features using a single camera, pp. 143–157.

Bensebaa, K. and Martins, M. P. (2008). Localization estimation for autonomous aerial navigation by matching images with different resolutions, *Proceedings of the 2nd WSEAS International Conference on Circuits, Systems, Signal and Telecommunications*, pp. 147–154.

Caballero, F., Merino, L., Ferruz, J. and Ollero, A. (2008). Vision-based odometry and slam for medium and high altitude flying uavs, *Journal of Intelligent and Robotic Systems* .

Campos, M. F. M. and de Souza Coelho, L. (1999). Autonomous dirigible navigation using visual tracking and pose estimation, *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* **4**: 2584–2589.

Dellaert, F., Fox, D., Burgard, W. and Thrun, S. (1999). Monte carlo localization for mobile robots, *Proceedings of the IEEE International Conference on Robotics and Automation*.

Elinas, P. and Little, J. J. (2005). σMCL: Monte-Carlo localization for mobile robots with stereo vision, *Proceedings of Robotics: Science and Systems*, Cambridge, USA, pp. 373–380.

GMA (2008). Google Maps API - Google Code., Disponível em: http://code.google.com/apis/maps/. Acesso em: 29 de outubro de 2008.

Iscold, P. (2007). Development of a Small Unmanned Aerial Vehicle for Aerial Reconaiscence, *International Congress of Mobility Engineering*, São Paulo, Brazil.

Karlsson, R., Schön, T., Törnqvist, D., Conte, G. and Gustafsson, F. (2008). Utilizing model structure for efficient simultaneous localization and mapping for a uav application, *Proceedings 2008 IEEE Aerospace Conference*, Big Sky, MT, USA.

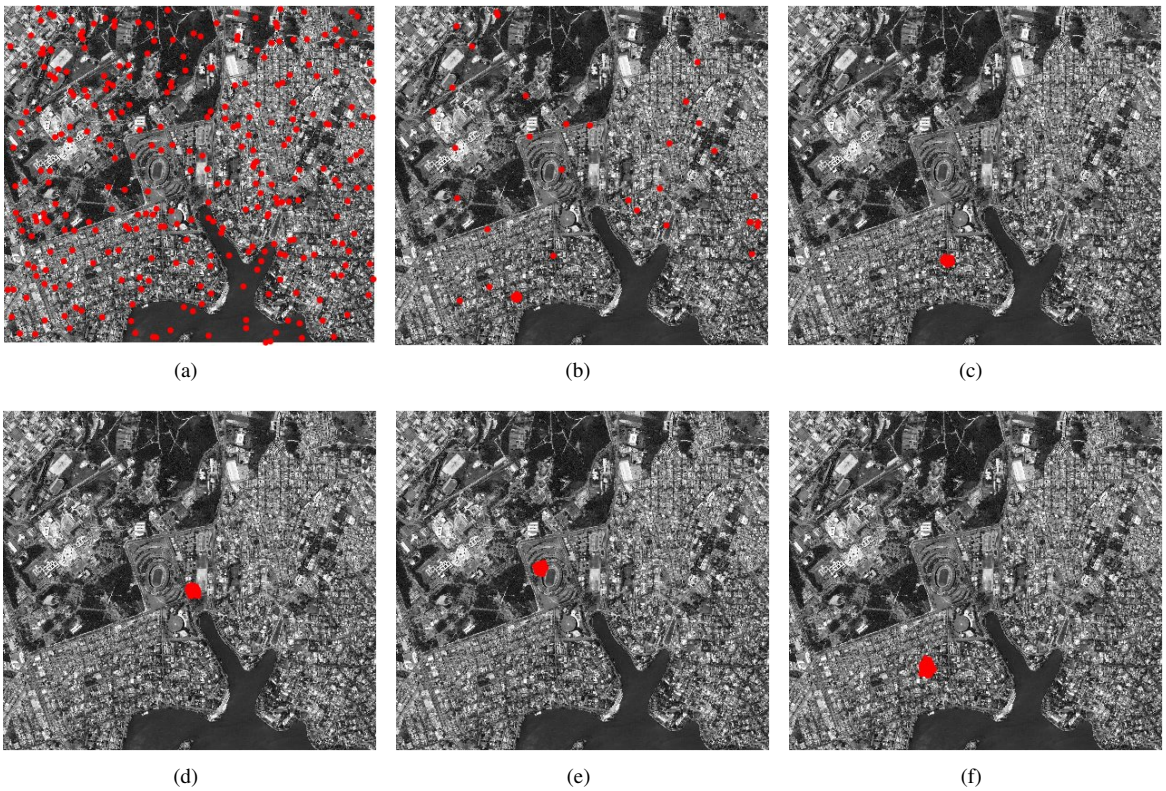

Figure 5: Particles' behavior during the trajectory

Kong, W., Egan, G. and Cornall, T. (2006). Feature based navigation for uavs, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 3539–3543.

Leung, K., Clark, C. and Huissoon, J. (2008). Localization in urban environments by matching ground level video images with an aerial image, *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* pp. 551–556.

Lowe, D. (1999). Object recognition from local scale-invariant features, *Proceedings of the Seventh IEEE International Conference on Computer Vision.* **2**: 1150–1157.

Ramisa, A., Tapus, A., de Mantaras, R. and Toledo, R. (2008). Mobile robot localization using panoramic vision and combinations of feature region detectors, *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* pp. 538–543.

Saripalli, S., Montgomery, J. F. and Sukhatme, G. S. (2002). Vision-based autonomous landing of an unmanned aerial vehicle, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2799–2804.

Se, S., Lowe, D. and Little, J. (2001). Vision-based mobile robot localization and mapping using scale-invariant features, *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2051–2058.

Sinopoli, B., Micheli, M., Donato, G. and Koo, T. (2001). Vision based navigation for an unmanned aerial vehicle, *Proceedings of the IEEE International Conference on Robotics and Automation* **2**: 1757–1764.

Steffen, R. and Forstner, W. (2008). On visual real time mapping for unmanned aerial vehicles, *ISPRS08*, p. B3a: 57 ff.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*, The MIT Press.

Wu, A. D. and Johnson, E. N. (2008). Methods for localization and mapping using vision and inertial sensors, *AIAA Guidance, Navigation and Control Conference and Exhibit*.

Zhang, N., Li, M. and Hong, B. (2006). Simultaneous localization and mapping using invariant natural features, *IEEE International Conference on Robotics and Biomimetics (ROBIO '06).*, pp. 1682–1687.

Zhang, W. and Kosecka, J. (2006). Image based localization in urban environments, *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, IEEE Computer Society, Washington, DC, USA, pp. 33–40.