3D Tracking and Control of UAV using Planar Faces and Monocular Camera

Manlio Barajas, José Pablo Dávalos-Viveros, and J.L. Gordillo

Center for Intelligent Systems, Tecnológico de Monterrey, Monterrey, México
{mf.barajas.phd.mty,a00805951,jlgordillo}@itesm.mx
http://www.itesm.edu

Abstract. A method for tracking the 3D pose and controlling an unmanned aerial vehicle (UAV) is presented. Planar faces of target vehicle are tracked using the Efficient Second Order Minimization algorithm, one at a time. Homography decomposition is used to recover the 3D pose of the textured planar face that is being tracked. Then, a cuboid model is used to estimate the homographies of the remaining faces. This allows switching faces as the object translates and rotates. Cascade and single PID controllers are used to control the aerial vehicle pose. Results confirm that this approach is effective for real-time aerial vehicle control using only one camera. This is a step towards an automatic 3D pose tracking system.

Keywords: 3D Tracking, 3D Pose Estimation, Aerial Vehicle Control, Homography Decomposition, Cuboid Tracking, Polygon Mesh Tracking

1 Introduction

For autonomous navigation, a localization method and a control strategy are determinant elements to achieve success in navigation tasks. Visual 3D pose estimation of objects in real environments has been an important topic in literature because cameras have shown to be a reliable source of environment information. Common visual approaches depend on depth information by means of stereo setups [16], [11] or laser range data [15].

In addition to cameras, modern UAV control approaches rely on multiple sensor data for pose estimation and control. Here we're interested in those approaches that work using image data as the main source of information. For example, in [14], [6], [10] and [12], visual servoing is done in order to control the vehicle pose.

3D object tracking can deliver 3D pose information to the vehicle controller. 3D Tracking may be accomplished by using full featured models [13], [4] or by using approximated models [7]. While full featured models allow having a very exact representation of the real world entity, in not all situations is well suited having any sort of known 3D model. Approximated models can be used to overcome this constraint. This was validated at theoretical level in [2].

In this article, the method presented in [2] for 3D pose tracking is validated with the control of an UAV. An aerial vehicle, which has 6 DoF, is manipulated using the feedback from the visual pose estimation method, in a position based visual servoing (PBVS) schema.

This article is divided as follows: section II will introduce work related to the problem of 3D tracking. On section III, a method for 3D object tracking is reviewed. Section IV presents the control strategy used to achieve control over the vehicle for then, in Section V, presenting the experimental results. In section VI conclusions are remarked.

2 Related Work

Related to this work, there are two main disciplines. First, 3D tracking, that is the base of the pose estimation that is used frame to frame to effectively control the aerial vehicle. But also, visual servoing, since output from the pose estimation is used to directly feed a control loop.

Visual servoing of aerial vehicles has been a well developed topic and because of its complexity, some works are focused on specific parts of the process. For example, in [6], a pure image based visual servoing (IBVS) for landing and take-off process is developed. While this work has the advantage of not requiring a full 3D reconstruction, it can't be used for full object control (executing paths). In a similar manner, in [10], a planar patch is used for 3D pose estimation of the UAV. This is more close to our work, but it's restricted in space. A more robust work is done in [14], where the full object pose is controlled. Unfortunately, this approach requires a stereo setup for 3D reconstruction. A common pattern in these works is the use of cameras on the UAV. In our work, we used a single camera out-of-board.

For 3D pose estimation and tracking, our work relies on homography decomposition for 3D reconstruction and standard 2D plane tracking. A close application of this is presented by Benhimane and Malis [8]. The Efficient Second Order Minimization algorithm is used for aligning 2D images over 8 parameters, for homography transforms. We also use this plane tracking algorithm in our method. But while Benhimane's work is focused on visual servoing using only one planar face, in this work we extend to multiple faces, so the object can freely rotate while the camera remains fixed.

There are other alternatives for 3D tracking in context of image alignment. The method presented by Cobzas and Sturm [5] uses standard 2D tracking to a 3D pose parameter space (6 parameters, one for each DoF). A drawback is that, when changing the parametrization to handle 3D pose changes, it results that even some movements that could be tracked without a problem by tracking planes individually, can produce failure when tracking them using this parametrization. Another idea introduced by them are constraints between planes to make tracking more robust. For our work, in the future we will use a similar constrained tracking for increasing robustness on a per plane basis.

On the same line of parametrizing directly in euclidean space, Panin and Knoll [13] proposed a method for tracking objects that uses Mutual Information as similarity measure, instead of the common SSD, and perform a Levenberg-Marquardt optimization. Authors report performance of 2 *fps* because of the Mutual Information step. Our approach, on the other hand works at real-time (more than 24 *fps*). Moreover, Panin's method requires at each iteration projecting a full CAD model to image plane (and a z-test for determining visibility). Our tracker just requires this projection once per registration, and can be simplified to a partial 3D reconstruction. A drawback from our method is that it will be limited to polygon meshes, but that is a desired feature towards automatic plane detection.

Another approach to 3D pose tracking is that of Manz et al. [7]. In their work, they use a custom simplified model for 3D pose tracking based on feature points. Their work shows robust performance on real-time vehicle tracking. But this method has the disadvantage of requiring a fine tuning of descriptors and the model for each different tracked target. While this provides robustness, it's not suitable for dynamic models.

One of the ideas behind this work is avoiding 3D representations of objects. Because monocular cameras are used, tracking in 3D involve at some point taking 2D image projections to 3D euclidean space. Handling 3D usually involves additional complexity (for example, building a 3D CAD model and rendering it). We have aimed at doing as much as possible on 2D space, and later interpreting the corresponding 3D representation.

3 3D Object Tracking using Planar Faces

In this section, a method for tracking objects that move in a 3D world is reviewed. This method uses planar faces to achieve full 3D object tracking. Only one face is tracked at image level using the ESM algorithm. Once a 2D homography between the reference template and the current image is obtained, homography decomposition is used to obtain a 3D reconstruction of that face. Then, a set of precomputed 3D transforms between the different faces of the model allow obtaining its full 3D reconstruction. A plane selection method is used to determine which face is the most convenient to track in the next iteration. If the face that is being tracked needs to be switched, then a 3D to 2D projection is done to obtain the current homography for the new face to be tracked. This steps are illustrated in figure 1. Please refer to [2] for more details on this method.

3.1 Plane Tracking of Planar Faces

In order to recover the full pose of the target object, visual tracking over a single face is done. Because faces are planar, it's possible to use existing plane tracking methods for image alignment. For this work, the Efficient Second Order Minimization proposed by Malis [9] was implemented. This method has proven

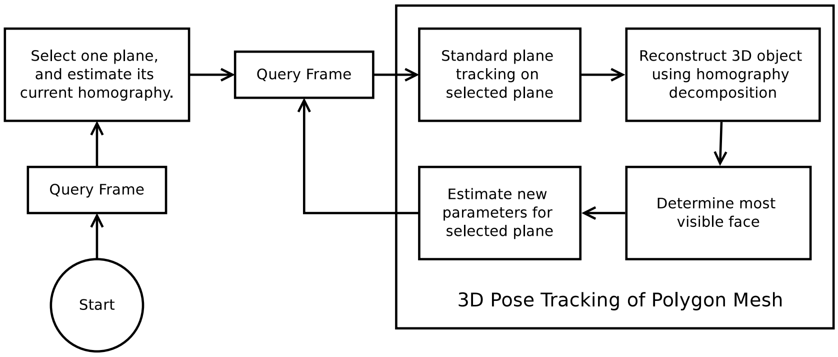
Select one plane, and estimate its current homography.
Query Frame
Standard plane tracking on selected plane
Reconstruct 3D object using homography decomposition
Query Frame
Estimate new parameters for selected plane
Determine most visible face
Start
3D Pose Tracking of Polygon Mesh

Fig. 1. General overview of 3D Object tracking method.

to provide higher convergence with less global error [9]. Also, this algorithm is fast enough to achieve real-time performance.

This method works by iteratively updating parameters $\mathbf{p} := \mathbf{p} \circ \mathbf{\Delta p}$ where $\mathbf{\Delta p}$ can be evaluated as:

$$\mathit{\Delta}\mathbf{p} \approx -2(\mathbf{J}(\mathbf{e}) + \mathbf{J}(\mathbf{p}_c))^{+}(\mathbf{s}(\mathbf{p}_c) - s(\mathbf{e})) \tag{1}$$

Where $\mathbf{p}$ are the current parameters, $\mathbf{J}$ is the Jacobian, as presented in [1], and $\mathbf{s}$ is the transformed current image. $\mathbf{e}$ is the identity parameter set, in this case $\mathbf{0}$. This method assumes the following homography parametrization:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \mathbf{G}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \\ p_7 & p_8 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2}$$

3.2 3D Object Reconstruction

The approach proposed in [2] is implemented to 3D reconstruct the object provided a single face of it and, in this case, assuming the real world dimensions of the object are known. In the 3D reconstruction stage, two main steps are done. First, a selected plane is tracked and then 3D reconstructed using homography decomposition. Then transformation matrices are used to obtain the full 3D object reconstruction.

Homography decomposition deals with reconstructing the 3D pose of a planar surface given its projection on image plane. When working with plane tracking, usually only 8 parameters are used to accomplish tracking, which are mapped to the 3 × 3 homography transform, as shown in equation 2.

In the pinhole camera model, the transformation matrix of extrinsic parameters $\mathbf{P} = [\mathbf{R} \mid \mathbf{t}]$ is a matrix composed of rotation $\mathbf{R}$ and translation $\mathbf{t}$. In order for $\mathbf{P}$ to be a valid transformation, $\mathbf{R}$ must have an orthonormal basis.

Using individual elements $h_{ij} : 1 \leq i, j \leq 3$ of $\mathbf{G}$, the following relations can be obtained:

$$\mathbf{P} = [\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}, \mathbf{t},] \tag{3}$$

$$\mathbf{P} = \begin{bmatrix} \frac{h_{11}-c_x r_{31}}{f_x} & \frac{h_{12}-c_x r_{32}}{f_x} & & \frac{h_{13}-c_x}{fx} \\ \alpha \frac{h_{21}-c_y r_{31}}{f_y}, & \alpha \frac{h_{22}-c_y r_{32}}{f_y}, & \mathbf{r}_1 \times \mathbf{r}_2, & \alpha \frac{h_{23}-c_y}{fy} \\ h_{31} & h_{32} & & 1 \end{bmatrix} \tag{4}$$

Where h_{ij} are the elements of homography matrix $\mathbf{G}$. c_x, c_y are camera center offset parameters. f_x and f_y are focal distances of the camera. Normalizing factor α must be estimated such that the magnitude of $\mathbf{r_1}$ and $\mathbf{r_2}$ be unitary. And advantage of this decomposition method is that it only requires one view, and camera intrinsic parameters.

Once the decomposition and 3D reconstruction of the tracked plane is obtained, the full object model may be reconstructed. For this work, the model is a cuboid. A cuboid has 6 planar faces and can be defined in terms of the planes that conform it. That is:

$$C = \{\mathbf{P}_i \mid 1 \leq i \leq 6\} \tag{5}$$

Where $\mathbf{P}_i$ refers to the homogeneous transformation matrix for plane π_i from its current referential to camera referential. For all faces in the cuboid C there exist at least one transformation matrix ${}^{i}\mathbf{T}_k$ that maps face referential k to face referential i. This can be expressed as:

$$\forall \pi_i \in C \; \exists \; {}^{i}\mathbf{T}_k \mid \mathbf{P}_i = \mathbf{T}_k \mathbf{P}_k \tag{6}$$

In the case of the cuboid used, these transforms can be estimated by assuming certain dimensions (up to scale factor) and then applying the corresponding rotations and translations. In fact, since there are 6 faces, a total of 30 transforms may be precomputed (5 for each of the 6 faces) so that it becomes straightforward obtaining the transform for face i from face k.

3.3 Face Selection

With the full 3D object reconstruction of the tracked object, the next step consist in determining which face is the most suitable for tracking in subsequent images. For this, the method outlined in [2] is used. This method works by finding the normal of the visible planes that is most aligned to the camera principal axis. In that article it's demonstrated that this is equivalent to finding face i such that:

$$\underset{i}{\operatorname{argmax}} \; l(i) = \{\mathbf{c} \cdot \mathbf{n}_i \mid \forall i \; : \; 1 \leq i \leq 6\} \tag{7}$$

Where $\mathbf{c}$ refers to the center of the tracked object relative to the camera coordinate system. $\mathbf{n}_i$ is a normal vector to plane i. It must be taken into account that the selected face will not always be the one with the most suitable position for tracking.

4 Visual Servoing of Aerial Vehicle

An *Ar.Drone Parrot* quadracopter was tracked and controlled using the presented approach. Since planar faces are required, a cuboid shaped object was adapted and installed on top of it (see figure 5). The 3D object tracking approach that was used allows obtaining a 4×4 transformation matrix that relates camera referential to object referential.

The objective of this research is having an aerial vehicle to execute a given trajectory, provided a set of 3D poses that are part of the path. In the case of the Ar.Drone vehicle, possible manipulations are: *roll*, *pitch*, *yaw* and *vertical speed*. It must be noted that since the vehicle may move freely in all directions, only set-points on translation are required. Yaw control can be used in order to have the vehicle's camera to look at certain places.

For the roll and pitch manipulated variables, a cascade PID control strategy was employed. This is required to have control over the speed at which the vehicle moves. The input is the desired x-y position. From the difference between the set-point and the measurement, a speed set-point is generated. A second controller takes this input and transforms it to a roll and pitch values. This controller is shown in figure 4.

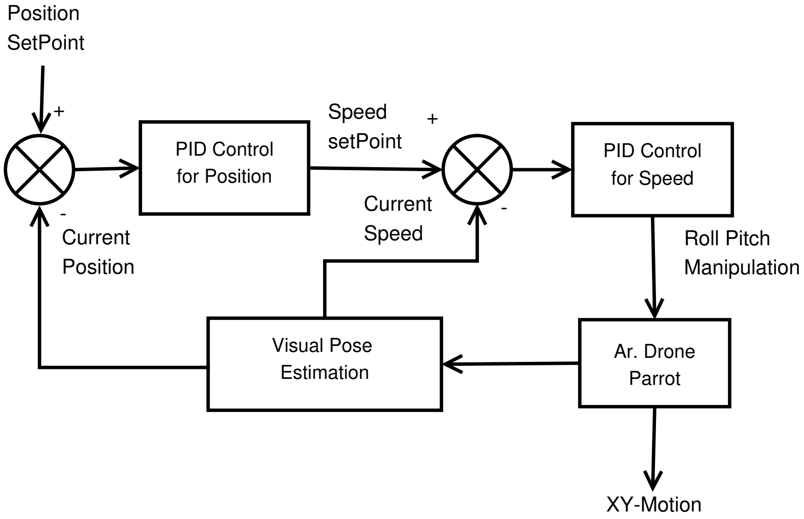

Fig. 2. Control loop for x-y position tracking. The first control loop controls the target speed of the drone. The second loop manipulates the pitch and roll to reach the target speed.

For the roll and pitch controller, the integral component of the speed controller plays a key role when perturbations are present, since in case of strong wind conditions, it will help to reduce the error by increasing the manipulation when the vehicle is having difficulty to move regardless of the pitch and roll manipulation.

For the yaw and vertical speed, simple proportional controllers were used, since built-in orientation and height control is good for allowing soft movements.

5 Experiments

Two exercises were run. First, the UAV was instructed to remain in “hover” state. Since there exist different perturbations on the environment (for example, the pressure effect produced by walls), the vehicle will not remain in that state by itself if no manipulation is applied. Results are shown in figure 3.

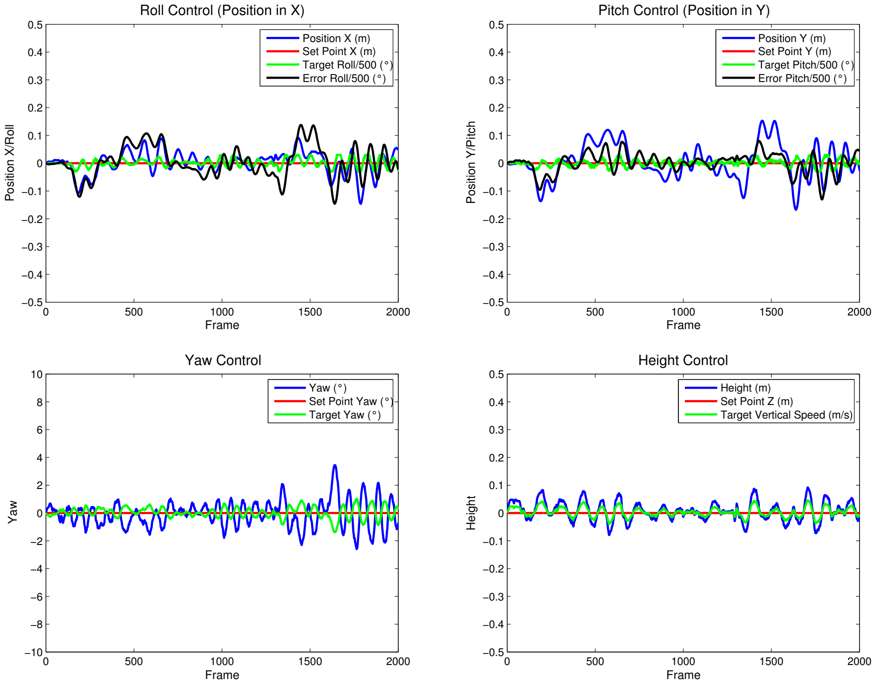

Fig. 3. Control variables and manipulation for the “hover exercise”. The drone should remain at a fixed position, but perturbations will affect the control. For this exercise, only one face was tracked

For the hover exercise, only one face of the cuboid is tracked, since perturbations are not capable to producing a face switch.

The second exercise consisted in a set of rotations over the yaw axis of the vehicle. Because the starting face is not visible at all stages of the test, the

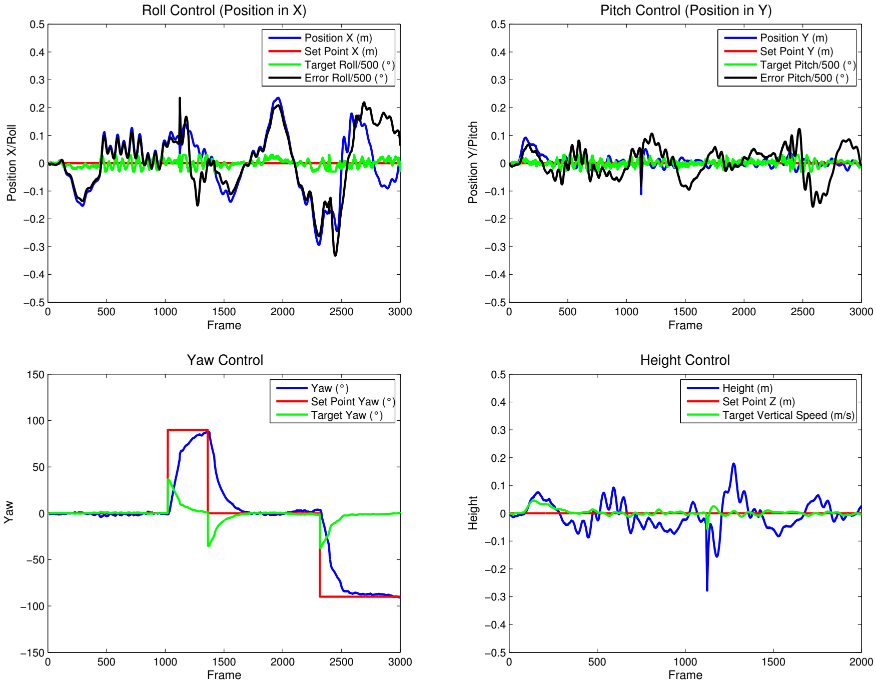

Fig. 4. Control variables and manipulation for the "rotation exercise". The drone should remain at a fixed position while the yaw should reach three provided set-points: 90°, 0° and −90°. It must be noted artifacts present in plots around frames 1220, 1460 and 2420. These are produced when tracked face is switched to a different face.

algorithm had to switch faces, consistently with employed face selection strategy. Figure 4 shows the result of this exercise.

Regarding 3D cuboid reconstruction and face selection, current results show that pose estimation presents substantial noise. Error in the reconstruction affects how the cuboid is projected from $\Re^3$ to $\Re^2$. The reconstruction error is related to the plane tracking process and the camera calibration used for homography decomposition.

Face selection could also be validated in the rotation experiment. In figure 5, that corresponds to exercise two, when the vehicle rotated, face selection algorithm selected the face that was more suitable for plane tracking. An implementation note is that the face selection should allow the tracker to completely converge, since using a non convergent solution can produce a race condition, where each time, a worst reconstruction produces a new face switch, and so on, until the tracker diverges.

Experiments confirm that our approach, that used only one out-of-board camera, is able to produce a stable control. Because the 3D position estimation method is new (especially, the face switching algorithm), it results difficult

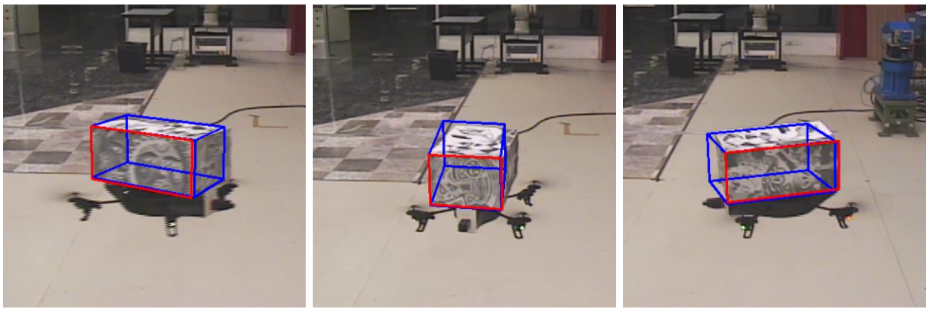

Fig. 5. Sample poses of the Parrot and the corresponding 3D reconstructions (wireframe). Selected face is marked with red. Reconstructions do not fit exactly to the object. Nevertheless, retro-projections of 3D transforms to image are close enough to be a good starting point for the first iteration of the tracker.

to compare these results to what is obtained by others, especially taking into account that vehicles are different in different environmental conditions.

6 Conclusions and Future Work

We have presented a method for visual servoing of a flying drone in a 3D environment. The performance of the method is completely real-time and capable of feeding a control loop that assumes no delay.

Even though the calibration did not produce 3D reconstructions such that, when projected back to image plane were a perfect fit for the pose of the cuboid object, the method was able to handle this small differences during face switching.

It also must be highlighted that only one camera was used. We showed how using only homography decomposition and simple transforms models (in $\Re^3$) it is possible to construct a 3D model of the target object.

An area of opportunity for this research is the fusion of data from drone IMU and other sensors with that of the vision. This could help reducing the noise in the pose estimation and face switching stage.

This work is aligned with our general objective of building automatic 3D trackers. Future work will include the connection of an automatic plane detection method with this approach for doing automatic object tracking.

Acknowledgement Authors want to acknowledge Consejo Nacional de Ciencia y Tecnología (CONACyT) and e-Robots Research Chair from Tecnológico de Monterrey, Monterrey for supporting this research.

References

1. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework: Part 1. International Journal of Computer Vision. 56, 221-255 (2004)
2. Barajas, M., Esparza, J., Gordillo, J.L.:Towards Automatic 3D Pose Tracking through Polygon Mesh Approximation. In: Advances in Artificial Intelligence IBERAMIA 2012. 531-540 (2012)
3. Bouchafa, S., Zavidovique, B.: Obstacle Detection "for free" in the C-velocity space. In: 14th IEEE International Conference on Intelligent Transportation Systems (ITSC). 308-313 (2011)
4. Brown, J.A., Capson, D.W.: A Framework for 3D Model-Based Visual Tracking Using a GPU-Accelerated Particle Filter. IEEE Transactions on Visualization and Computer Graphics. 18, 68-80 (2011)
5. Cobzas, D., Sturm, P.: 3D SSD Tracking with Estimated 3D Planes. In: The 2nd Canadian Conference on Computer and Robot Vision. 129-134 (2005)
6. Daewon, Lee. Ryan, T. Kim, H.J.: Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. In; IEEE International Conference on Robotics and Automation (ICRA). 971-976 (2012)
7. Manz, M., Luettel, T.: Monocular model-based 3D vehicle tracking for autonomous vehicles in unstructured environment. In: IEEE International Conference on Robotics and Automation (ICRA). 2465-2471 (2011)
8. Benhimane, S., Malis, E.: Homography-based 2D Visual Tracking and Servoing. The International Journal of Robotics Research. 661-676 (2007)
9. Malis, E.: Improving vision-based control using efficient second-order minimization techniques. In: IEEE International Conference on Robotics and Automation (ICRA). 1843-1848 (2004)
10. Mondragón, I.F. Campoy, P.. Martinez, C. Olivares-Méndez, M.A.: 3D pose estimation based on planar object tracking for UAVs control. In: IEEE International Conference on Robotics and Automation (ICRA). 35-41 (2010)
11. Munozsalinas, R., Aguirre, E., Garciasilvente, M.: People detection and tracking using stereo vision and color. Image and Vision Computing. 25, 995-1007 (2007)
12. Ozawa, R. Chaumette, F.: Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach. In: IEEE International Conference on Robotics and Automation (ICRA). 5670-5676 (2011)
13. Panin, G., Knoll, A.: Mutual Information-Based 3D Object Tracking. International Journal of Computer Vision. 78, 107-118 (2007)
14. Salazar, S., Romero, H., Gomez, J., Lozano, R.: Real-time stereo visual servoing control of an UAV having eight-rotors. In: 6th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). 1-11 (2009)
15. Xiaowei Shao, Huijing Zhao, Nakamura, K., Katabira, K., Shibasaki, R., Nakagawa, Y.: Detection and tracking of multiple pedestrians by using laser range scanners. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2174-2179 (2007)
16. Tonko, M., Nagel, Hans-Hellmut: Model-Based Stereo-Tracking of Non-Polyhedral Objects for Automatic Disassembly Experiments. International Journal of Computer Vision. 37, 99-118 (2000)