https://doi.org/10.17979/spudc.9788497498081.1124

Accuracy analysis of marker-based 3D visual localization

Alberto López-Cerón
Universidad Rey Juan Carlos, alberto.lopezceron@gmail.com
José M. Cañas
Universidad Rey Juan Carlos, josemaria.plaza@urjc.es

Abstract

3D localization from images is an useful capability for robots and cameras. One successful approach is to rely on visual SLAM techniques. Another approach, maybe more robust, is to use visual markers in the environment. In this paper a study about the accuracy of marker based visual 3D localization is presented, using *AprilTags* markers and the *solvepnp* algorithm in OpenCV library. The impact of distance to markers, number of markers, their position in the image on accuracy of the 3D estimated pose is experimentally measured and analyzed.

1 Introduction

Cameras are ubiquitous sensors: robots, drones, mobile phones, etc. are typically endowed with one or more. One useful piece of information that can be extracted from images is the 3D localization of the camera. There are many applications where the 3D visual localization is extremely useful. For instance, augmented reality applications in order to calculate how virtual objects should be located and oriented in the images. Also in robotics, like the self localization of an autonomous industrial robot, to generate ground-truth robot trajectories and close control loops or to estimate the relative position of tagged objects for a humanoid robot (like Atlas from Boston Dynamics).

This problem has been addressed from several fields like robotics, augmented reality and computer vision. Many techniques have appeared like probabilistic visual self-localization algorithms (particle filters...), visual odometry (image registration...), etc. In recent years, visual SLAM techniques like monoSLAM, PTAM, SVO, etc. have been very successful. Another approach is based on visual markers, whose position is known in advance. These fiducial systems provide camera-relative position and orientation of a tag and such estimation is known as the Perspective-n-Point (PnP) problem [10, 7].

Most of these techniques work with color images. In the last five years RGBD sensors have appeared and simplified the problem, helping to estimate the scale of the estimations. For instance the recent project Tango[1] from Google uses both color and depth images.

Operation in real time, robustness, re-localization capability and accuracy are very desirable features of the visual 3D localization algorithm. The goal of the paper is to study the limits of the standard solution to PnP problem and its accuracy estimating the position and orientation of the camera from the markers.

Second section of this paper presents previous works on mark-based visual 3D localization. Third one describes our implementation of the classic solve-PnP algorithm extending it to work continuously and with several markers at the same time. Experiments section presents our accuracy analysis and the conclusions end the paper.

2 Related works

Several types of markers have been explored in the literature, many of them closely related to Augmented Reality applications. ARToolkit [6, 11] and ARToolkitPlus use tags contained in a square-shaped payload surrounded by a black border. ARToolkit is now open source. Its payload was not directly encoded in binary. ARToolkitPlus was succeeded by Studierstube Tracker [2], closely oriented to mobile phones.

ARTag [4] is a bitonal system of markers consisting of a square border and an interior region filled with a 6x6 of black and white cells. More recent proposals are CALTag [2], with high precision markers oriented to camera calibration; RUNEtag [3], oriented to high resilience to occlusions; and ARUCO [5], whose code has been integrated as a module in OpenCV library.

Interesting localization accuracy analysis can be found at [9] and [1], the last one using ARToolkit

[1] https://www.google.com/atap/project-tango
[2] http://handheldar.icg.tugraz.at/stbtracker.php

markers.

3 Marker-based 3D visual localization

Many of the approaches to localization and navigation of robots in the last years have been based on visual markers, mainly because they are cheap and not extremely hard to computationally detect if they are well selected. To this aim, their main distinctive features are the contrast and the shape: the first one has to be as high as possible and the second one must be considerably different from the rest of the nearby objects. As an example, the AprilTags or ArUco markers can be cited, which are similar to the QR codes, but designed to store less information in a more robust manner. For this work, the AprilTags library [8] has been used [3], in particular the C++ implementation [4], which is open source licensed.

Figure 1: Marker set from AprilTags

3.1 Detection of markers in image

AprilTags is a 2D marker detection system that describes a robust method to find the markers in the image and proposes a precise segmentation algorithm. On the other hand, it describes a coding system that deals with specific problems of the 2D bar code systems: robustness to rotation and robustness to false positives arising from natural imagery.

The first main component of the system, the marker detector, is designed to have a very low false negative rate, so its false positive rate is high. That is why it relies on the second main component, the coding system, to reduce this rate to an acceptable level. This last component can generate codes for any marker size and minimum Hamming distance. Its approach explicitly assures the minimum Hamming distance for the four rotations of each marker and discards the markers of low geometry complexity.

Making use of the commented algorithm, the C++ library provides, for each image passed, the position in image coordinates of the four corners of every detected marker.

[3] http://april.eecs.umich.edu/wiki/index.php/AprilTags
[4] http://people.csail.mit.edu/kaess/apriltags

3.2 3D information from a marker

The PnP problem is one of the classic problems in computer vision and photogrammetry. The estimation of the position and orientation based on points of correspondence has been intensely studied in the last decades and is essential in numerous fields of application. Such problem could be formally stated in the following way: given a set of matches between n reference 3D points and their projections in the image, find the position and the orientation of the calibrated camera with respect to those control points. Namely, what is to be determined is the rotation-translation matrix that transfers the coordinate system of the world to the image one.

There are basically two types of methods to solve the problem in the case $n < 6$: closed form methods (that convert the problem in a polynomial equation) and optimization iterative methods (that try to solve it by the minimization of a cost function properly defined). In this study an iterative one has been selected, making use of the OpenCV library, that provides a function (solvepnp) which accepts as arguments the control points (the four corners of a marker with respect to its center), their projections (given by the AprilTags library) and the intrinsic parameters of the camera (which has been previously calibrated). As a result it returns the estimated pose of the marker with respect to the camera, in the form of a rotation vector (in the Rodrigues format) and a translation vector. The cost function used by the method of this function is the re-projection error, which is the sum of the squares of the distances between the provided projections and those calculated with the corresponding solution. The great strength of these kind of methods relies on that they are usually extremely fast and accurate. As a drawback, they can only find a feasible solution each time (when $n < 6$ the uniqueness of the solution can not be guaranteed).

From the obtained translation and rotation vectors, the corresponding rotation-translation matrix can be formed ($RT_{CameraMarker}$), with the help of the OpenCV function called Rodrigues. Then, to change to the reference system of the marker, the inverse matrix is calculated: $RT_{MarkerCamera}$. Finally, to get the pose of the camera with respect to the world, the following matrix product is performed:

$$RT_{WorldCamera} = RT_{WorldMarker} \cdot RT_{MarkerCamera}$$

3.3 Fused 3D estimation

More than one marker may appear in an image so it is useful to fuse all the individual estimations from each marker, hopefully improving the robustness of the final estimation. The 3D fusion performed is a weighted average of the coordinates and angles of all the estimated poses: the closer the marker is, the bigger is the weight assigned. This fusion is done in every received image, so an estimation of the absolute pose of the camera is continuously available.

$$ratio_i = \frac{weight_i}{weight_{total}}$$

This calculation is straightforward for position coordinates:

$$[x, y, z]_{fusion} = \sum([x_i, y_i, z_i] \cdot ratio_i)$$

but needs a careful management for angles because of their circular nature. To deal with them, the arctangent of the sum of the sines of the corresponding angle is calculated, divided by the sum of the cosines.

$$\alpha_{fusion} = atan(\frac{\sum(sen(\alpha_i) \cdot ratio_i)}{\sum(cos(\alpha_i) \cdot ratio_i)})$$

4 Experiments

In order to study the accuracy of this 3D visual localization algorithm, a camera has been placed close to a visual marker in different relative positions and orientations. Then the mean distance error and the mean angular error have been measured. The *distance error* is the euclidean distance between the true position and the estimated one. The angular error is an average of the errors in yaw, pitch and roll, taking into account the circular nature of the angles. The experiments have been performed within the standard robotics simulator Gazebo, which provides the true camera pose information. Some of them were also carried with a real camera, a Logitech WebCam Pro9000 at 25fps with 640x480 frames, that was properly calibrated with OpenCV tools.

The algorithm delivers the relative pose of the camera according to the detected marker, which has its own coordinate frame (Figure 2). Then, it is transformed to the absolute pose of the camera taking into account the absolute pose of each marker in the world. For the experiments, *yaw* is considered as the rotation of the camera around the marker Z axis, *pitch* the rotation of the camera around the marker Y axis and *roll* the rotation of the camera around the marker X axis.

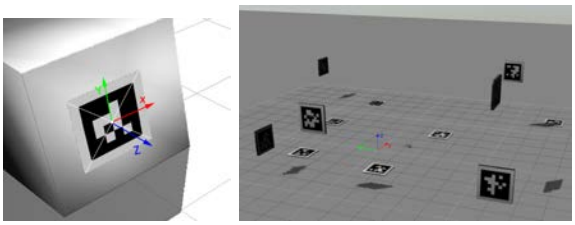

Figure 2: Relative-to-marker and absolute coordinate systems

4.1 Effect of yaw and distance

In this first experiment only one marker has been used, which was observed by the camera in the center of the image.

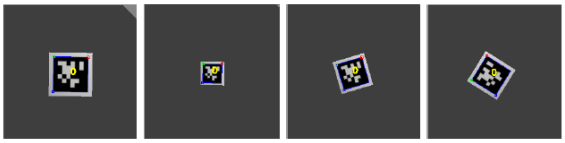

Figure 3: One marker at two different distances and angles

Figure 4 shows the radial and the angular errors depending on the distance and the *yaw* angle at the same time. The first interesting point is how the estimation degrades once a certain distance is passed (about 4 meters), which is true for both radial and angular error and independently from the orientation. The error increases with the distance, smoothly from 1 to 4 meters (the mean radial error increases with the distance, but remains below 10 cm and the mean angular error below 0,02°), until the estimation gets completely degraded beyond 4 m.

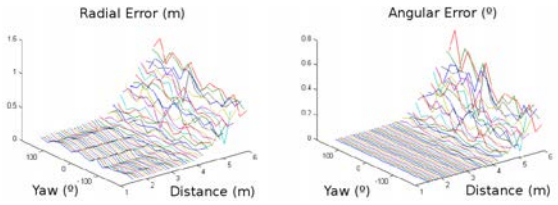

Figure 4: Errors in front of distance and yaw, 1 marker

The dependence with yaw is lower, getting similar values in the whole range of angles (+-180° because in this case the camera can make a complete turn and the marker is detected the whole time).

The experiments in real environment lead to similar conclusions, as it can be observed in figures

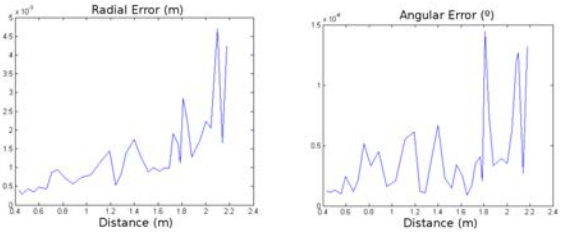

Figure 5: Errors in front of distance, real setting

5 and 6. The error increase with distance is also observed, as it is the low influence of yaw.

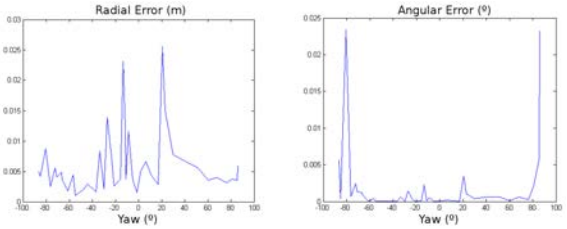

Figure 6: Errors in front of yaw, real setting

4.2 Effect of pitch and distance

In this experiment the focus is how the estimation behaves if the distance and pitch between the camera and the marker varies. Increasing or decreasing this angle from 0º makes the camera capture the marker more and more heeled over. The camera was not moved just the marker's orientation, because this way the desired angle was controlled easier.

Figure 7: One marker at different pitch angles

Figure 8 shows a noticeable dependence with the distance, as well as in the previous experiment. Nevertheless, there is a difference, the error increase is not uniform in the whole range of pitch. In addition, the error is bigger when the pitch is small, that is, when the parallelism between the marker and image planes is high. Moreover, having a certain pitch between the marker and the camera attenuates the effect of the distance, getting better estimations than in the parallel case at the same distance. So, it can be concluded that, at the time of choosing the orientation of the markers to use in a real system, a certain pitch is advisable, as well as avoiding the camera to observe the markers totally parallel to its image plane.

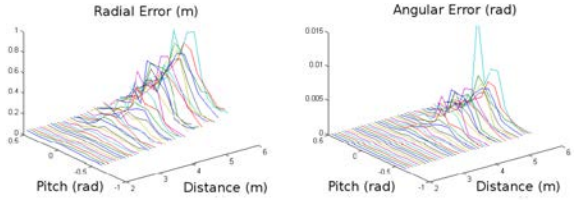

Figure 8: Errors in front of distance and pitch

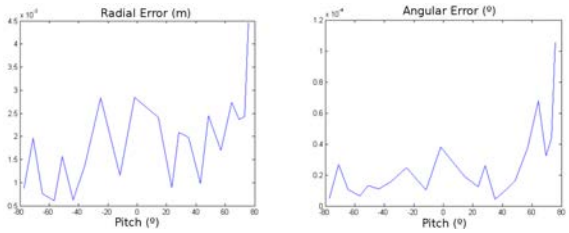

Figure 9: Errors in front of pitch, real setting

In the real setting the error decrease trend is partially observed, though not completely most likely due to the inaccuracy of the real pose measurement method.

4.3 Effect of roll and distance

In this experiment the focus is to study the dependence of the estimation with the roll angle. It is equivalent to the one in the previous section: a marker has been included in the Gazebo virtual environment and its orientation was modified so the camera saw it with different roll angles, that is, with different inclinations (with a 0 roll the marker is parallel to the image plane).

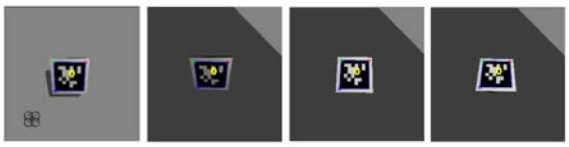

Figure 10: One marker at different roll angles

Figures 10 and 11 shows something similar to the pitch case is: the error increases with distance and this distance does not affect that much if there is a certain roll. Again, the conclusion is that a certain roll is advisable for a better quality of the estimation, instead of setting the markers parallel to the image plane of the camera.

In the real setting there are no measurements for positive values of roll because it is difficult to incline the marker without falling down. In that

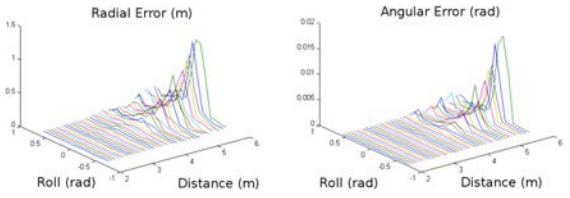

Figure 11: Errors in front of distance and roll

range the expected trend is first observed while the absolute value of the roll increases, but a peak appears later, that may be caused by a puntual wrong marker detection.

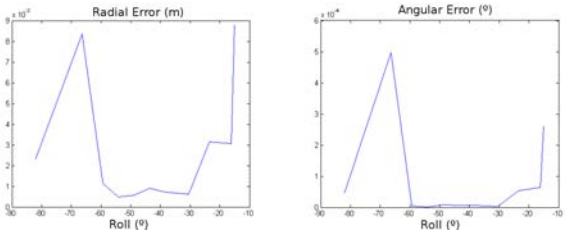

Figure 12: Errors in front of roll, real setting

4.4 Number of markers

Another interesting issue is knowing if the estimation gets better making use of more markers. First, two of them have been used in a diagonal disposition, so the camera observes them at the corners of the image when it is close to them.

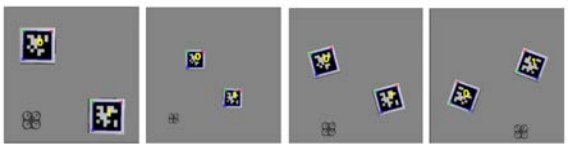

Figure 13: Two markers at two different distances and angles

Analyzing the figure 14, the errors increase with the distance and there is low dependence on the angle, similar to previous experiments, but new interesting information arises: first, the error size has decreased in both radial and angular errors and, second, the distance where the degradation gap occurs is bigger. The average radial error is around 5 cm until 4 m of distance and is no bigger than 10 cm until 5 m (with a single marker the mean error was 10 cm until 4 m and it went up to 40 cm at 5 m). On the other side, the mean angular error is no bigger than 0,02º until 5 m of distance. That is, with only one marker the estimation was considerably degraded after 4 m

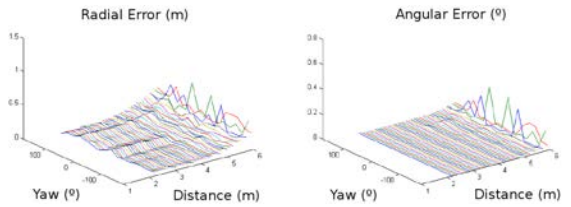

Figure 14: Errors in front of distance and yaw, 2 markers

while, with two markers, the error at 5 m is still acceptable.

In another experiment four markers have been used (Figures 15 and 16), all of them in the same plane and with their borders parallel. Again something similar is observed, with an even smaller error and with a smoother degradation gap. The improvement with respect the two markers case is specially noticeable after 4,5 m, because before that distance the error values are similar. At a distance of 5 meters the radial error in this case is around 8 cm (in the previous case it began to exceed 10 cm), while the mean angular error is of 0,01º (with two markers it was around 0,02º).

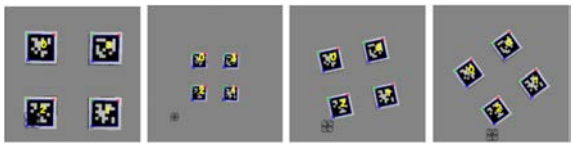

Figure 15: Four markers at two different distances and angles

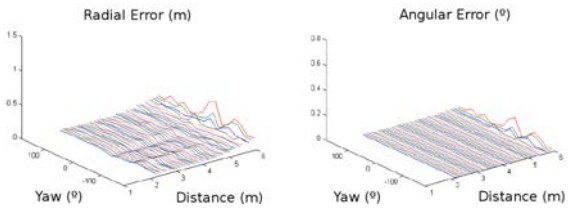

Figure 16: Errors in front of distance and yaw, 4 markers

In addition to the error representation depending on the angle and the distance at the same time, the relationship between the error and each parameter has been analysed independently. For example, the Figure 17 shows the evolution of the error depending on the distance alone, without taking into consideration the angle between the camera and the markers. For that, the average error in the whole yaw range has been calculated.

In these graphics, two of the observations made before are even more clear: the error increases

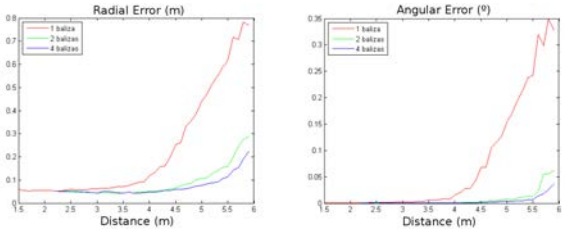

Figure 17: Errors in front of distance

with the distance and decreases with the number of markers. This last point is also noticeable in the graphics of error in front of yaw, where for each yaw value the mean error in the whole range of distances has been calculated.

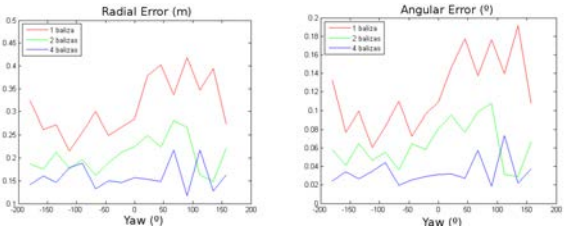

Figure 18: Errors in front of yaw

4.5 Markers in different planes

In this experiment the effect of an additional marker, perpendicular to the plane of the others, on the estimation was studied.

First, the four markers case have been measured again, but this time they were not parallel to the floor, because then the additional marker would have been completely perpendicular to the camera. Comparing figures 20 and 16, both corresponding to a case with four markers, but inclining in a case and parallel to the floor in the other, an improvement in bigger distances can be observed, having no peaks of error. Again, the inclination is a good factor for improving the estimation. An acceptable response in the whole range of angles and distances studied is observed, exceeding the mean radial error 10 cm at a distance of 6 meters, keeping the angular error below 0,001°.

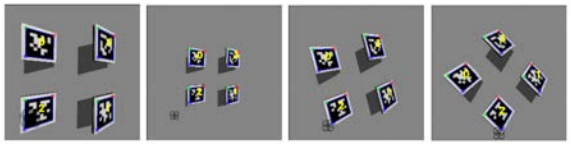

Figure 19: Four inclined markers at two different distances and angles

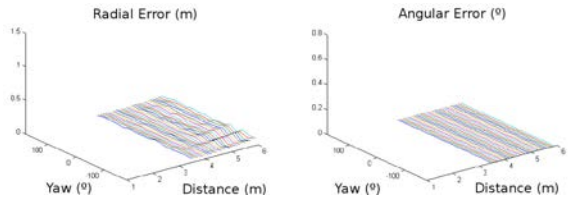

Figure 20: Errors in front of distance and yaw, 4 inclined markers

Second, a fifth marker has been added and the experiment has been repeated. Initially, a similar behavior is observed if Figures 20 and 22 are compared.

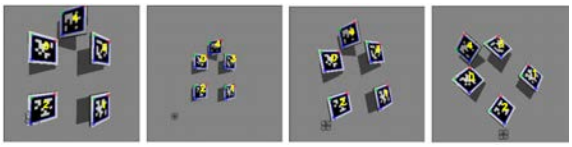

Figure 21: Four markers and a perpendicular one at two different distances and angles

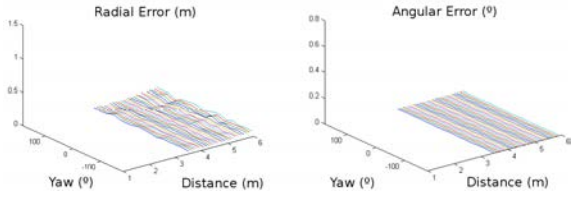

Figure 22: Errors in front of distance and yaw, 4 markers and 1 perpendicular

To better appreciate the differences, the following figures are presented, which directly compare the results obtained in both cases, but for each parameter separately. Having a look at the radial error comparatives (left part of figures 23 and 24): the error is lower using the fifth value, which is observed in front of distance and in front of yaw too.

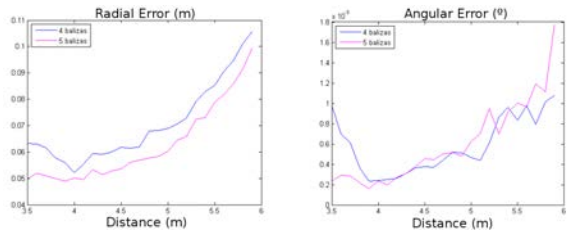

Figure 23: Errors in front of distance, 4 and 5 markers

The angular error (right part of figures 23 and 24) behaves in a weird way. In front of distance, the

angular error with five markers begins below, but after 5 m this trend seems to revert surprisingly. In front of yaw (figure 24 right) takes values very similar in both cases, except in the range from 0º to 100º, where the value for four markers present a considerable peak. This may be due to an exceptional bad corner detection of one or more markers.

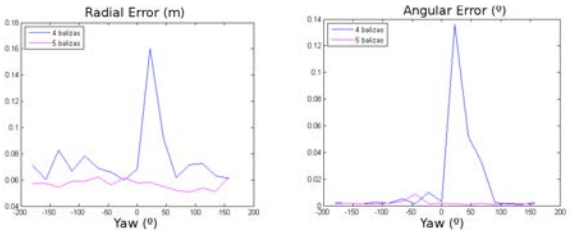

Figure 24: Errors in front of yaw, 4 and 5 markers

Another aspect to consider when a perpendicular marker is included is to separate the radial error in error in XY and error in Z.

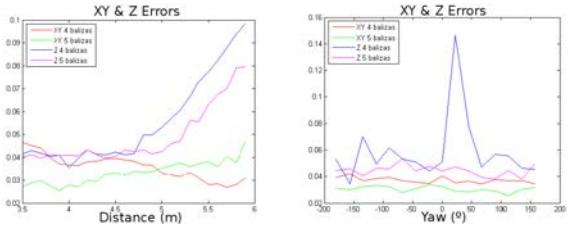

Figure 25: Error XY and Error Z in front of distance and yaw, 4 and 5 markers

Left part of Figure 25 shows that what affects the most to the radial error increase with distance is the error in Z, not in XY. The error in XY remains around 4 cm in the whole range of studied distances, while the error in Z begins to substantially increase after 4,5 m. In addition, it is observed how the inclusion of the fifth marker affects positively to the error in Z, being lower at bigger distances.

With respect to the errors in front of yaw, presented in the right part of figure 25, it is also observed the improvement with the perpendicular marker, being the errors (both XY and Z) smaller in the whole range of studied distances.

4.6 Pattern position in image

These experiments aim to study if the distance of the marker to the center of the image affects the quality of the estimation. To do so, the image has been divided in 9 zones and the camera has been positioned so the marker was captured in each one. Twenty measurements have been performed each time.

Figure 26: Marker in 3 of the 9 possible positions

The table 1 shows the radial error measured in each case. As it can be observed the differences are minimal, being the biggest one of 7,1 mm. It is not observed either trends in the error increase or decrease that may lead to think that the estimation is better when the marker is centered or in a corner of the image.

Radial error (cm)		
10,93	10,63	10,65
10,45	10,53	10,64
10,25	10,62	10,96

Table 1: Radial error in image translation

In all these 9 zones of the image the XY and Z error were also studied separately. In the left part of the table **??** a significant trend may be observed: the zones of the image where the XY error is lower are the corners, which is related with the previous conclusions that indicated that a certain roll and pitch (not being the marker completely centered) increases the quality of the estimation. However, because this error is much lower than the Z error and being this independent from the translation, the trend gets masked within the global radial error.

XY error (mm)			Z error (cm)		
2,95	11,28	4,98	10,92	10,56	10,63
9,37	6,28	7,40	10,41	10,51	10,61
4,09	5,09	2,13	10,24	10,60	10,96

Table 2: Radial error in image translation

5 Conclusions

The main conclusions that can be drawn from the experiments performed are the following. First, there is a clear error dependence (both radial and angular) on the distance to the markers. Under 4 m the distance error keeps below 5 cm. The further a marker is from the camera, the bigger the error in the 3D estimations. This can be mitigated

if more markers are included in the scene, increasing the maximum valid distance under which the estimation error is reasonably small.

Second, using perpendicular markers (markers in several planes) improves the accuracy, mainly in Z. Using markers with a certain inclination in roll or pitch improves the accuracy of the estimations over using only markers in a plane parallel to the image plane.

Third, the impact of the yaw angle between the marker and the camera on the quality of the estimation is small.

And finally, the trends observed in the virtual experiments have also been noted in the real world experiments, although they are not completely equivalent, probably due to the inaccuracy of the method used to estimate the real true pose.

Acknowledgements

This research has been partially sponsored by the Community of Madrid through the RoboCity2030-III project (S2013/MIT-2748), by the Spanish Ministerio de Economía y Competitividad through the SIRMAVED project (DPI2013-40534-R) and by the URJC-BancoSantander.

References

[1] D.F. Abawi, Bienwald J., and R. Dorner. Accuracy in optical tracking with fiducial markers: an accuracy function for artoolkit. In IEEE Computer Society, editor, *ISMAR'04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 260–261, november 2004.

[2] Bradley Atcheson, Felix Heide, and Wolfgang Heidrich. CALTag: High precision fiducial markers for camera calibration. In Christof Rezk-Salama (Eds.) Reinhard Koch, Andreas Kolb, editor, 15th *International Workshop on Vision, Modeling and Visualization*, Siegen, Germany, November 2010.

[3] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello. Rune-tag: A high accuracy fiducial marker with strong occlusion resilience. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 113–120, Washington, DC, USA, 2011. IEEE Computer Society.

[4] M. Fiala. Artag, a fiducial marker system using digital techniques. In IEEE Computer Society, editor, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596, Washington, DC, USA, 2005.

[5] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[6] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In IEEE Computer Society, editor, *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pages 85–95, Washington, DC, USA, 1999.

[7] V. Lepetit, Moreno-Noguer F., and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *Int. Journal Computer Vision*, 81:155–166, 2009.

[8] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation ICRA-2011*, pages 3400–3407, Sanghai, may 2011.

[9] Katharina Pentenrieder, Peter Meier, and Gudrun Klinker. Analysis of tracking accuracy for single-camera square-marker-based tracking. In *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, Koblenz, Germany, September 2006.

[10] Li Shiqi, Xu Chi, and Xie Ming. A robust o(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Anaylisis and Machine Intelligence*, 34(7):1444–1450, jul 2012.

[11] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In IEEE Computer Society, editor, *ISMAR'08: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 125–134, Washington, DC, USA, 2008.