Chapter
Precise Localization for Aerial Inspection Using Augmented Reality Markers

A. Amor-Martinez, A. Ruiz, F. Moreno-Noguer and A. Sanfeliu

Abstract This chapter is devoted to explaining a method for precise localization using augmented reality markers. This method can achieve precision of less of 5mm in position at a distance of 0.7m, using a visual mark of 17mm x 17mm, and it can be used by controller when the aerial robot is doing a manipulation task. The localization method is based on optimizing the alignment of deformable contours from textureless images working from the raw vertexes of the observed contour. The algorithm optimizes the alignment of the XOR area computed by means of computer graphics clipping techniques. The method can run at 25 frames per second.

1 Introduction

In order to achieve high precision in the localization of UAVs for manipulation purposes (e.g. grasping, insertion, etc.), one way is to use visual markers. These methods, ARToolkit [5], can obtain the pose of any object with high precision using planar visual markers. They are prepared to cope with light changes, partial occlusion and inter-marker detection, obtaining accurate results [2]. When working with textured images, a very powerful registration method is Lucas-Kanade (LK) [8]. In case of planar contours, the homography methods [9][4] are also excellent methods, but they typically require complex optimization schemes. However, these methods can not obtain a precise pose when using non planar visual marks.

In this section we will explain a method published in [1] that can work with slow CPUs, low resolution cameras and small image deformations. The method consists of shape registration from extracted contours in an image. Instead of working with dense image patches or corresponding image features, the method optimize a geometric alignment cost computed directly from the raw polygonal representations of the observed regions using efficient clipping algorithms. Moreover, instead of doing 2D image processing operations, the optimization is performed in the polygon representation space, allowing real-time projective matching. Deformation modes are easily included in the optimization scheme, allowing, for example, accurate regis-

Table 1 Symbols for contour matching with deformation modes.

Definition	Symbol
Transformation model: $\mathbf{x}' = W_{\mathbf{p}}(\mathbf{x})$	$\mathbf{x}'$
Parameters for the image transformation and deformation modes	p and α
Observed contour	O
Template to match, and template as function of deformation parameters	T and $\mathbf{T}()$
Observed shape	I
Residual vector	$\mathbf{f}$
Jacobian $J = \partial\mathbf{f}/\partial\mathbf{p}$	J
Update rule	$\Delta\mathbf{p}$
Gradient	∇C
Hessian	$\mathbf{H}$
Local deformation required to improve alignment	$\delta_{\mathbf{p}}(\mathbf{x})$
Segment joining nodes k and k+1	S_k

tration of different markers attached to curved surfaces using a single deformable prototype. As a result, the method achieves very good object pose estimation precision in real-time, which is very important for interactive UAV tasks, for example for short distance surveillance or bar assembly. The method achieves very good precision, with an average error of less than 5mm in position at a distance of 0.7m, using a visual mark of 17mm x 17mm.

We have developed an efficient registration method for contours which consists on a Gauss-Newton optimization of a natural geometric alignment cost based on polygonal XOR clipping. The method is based on the whole image, without correspondences and noise tolerant, while working directly on a simple polygonal representation of region boundaries. All necessary optimization magnitudes (gradient and Hessian) are computed in closed form from vertexes coordinates.

The method is very precise and can compete against vision-based global positioning and motion capture systems as a low cost on-board solution for small object pose estimation (see Fig. 1) [1]. Also, these systems are not really appropriate for outdoors, where the environment is less controllable in most situations.

2 Contour-based registration

For simplicity we assume that the regions of interest are represented by piecewise linear contours obtained from standard image processing functions for thresholding, contour extraction, and polygon reduction. A natural alignment cost not based on explicit landmarks or correspondences is the total area of discrepancy between target and transformed template. The error regions can be efficiently obtained by means of an XOR (symmetric difference) clipping algorithm working on region boundaries [3], and their areas can be easily computed just from the contour nodes.

[1] All the figures in this section are from the authors and have already appeared in proceedings of IEEE ICRA in [1].

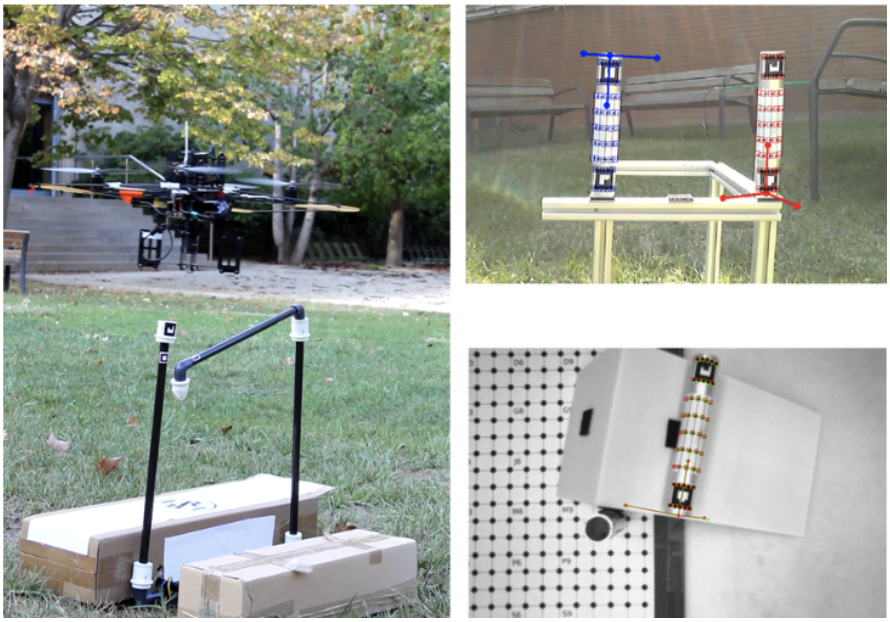

Fig. 1 Left: Case scenario we consider in this paper of a quadrotor under a supervision task. Right: Images of the bars acquired with the onboard cameras. Our goal is to recover the pose of the bar from the squared markers at the opposite sides of the bars. This kind of markers can be easily deployed in any kind of surface. Note, however, that the difficulty of estimating pose from these marks is specially difficult due to their small size. Dotted patterns are just used for ground truth computation, and are not used by our algorithm.

Given a transformation model $\mathbf{x}' = W_{\mathbf{p}}(\mathbf{x})$, the contour registration problem will be formulated as finding the parameters $\mathbf{p}$ that minimize the error area XOR(O, $W_{\mathbf{p}}$ (T)) for the observed contour O and template T. This can be solved using Gauss-Newton's iterative optimization: Given a residual vector $\mathbf{f}$ with Jacobian $J = \partial\mathbf{f}/\partial\mathbf{p}$, the squared error $C = 1/2\mathbf{f}^{\mathbf{T}}\mathbf{f}$ can be reduced by using the update rule $\Delta\mathbf{p} = -H\mathrm{i}\nabla C$, where $\nabla C = J^T\mathbf{f}$ and the Hessian is approximated by $\mathbf{H} = J^T J$. Exact residuals for contour alignment would require explicit template-observation correspondences, which are assumed not available. For efficiency and simplicity we will work just with the XOR error regions, without any further image or contour processing steps.

We propose a variant of Gauss-Newton with an infinite, continuous vector of approximate residuals for all points in the contour. These residuals and the required optimization magnitudes are efficiently computed in closed form from the nodes of the XOR error polygons. Each point in the contour produces two residuals in $\mathbf{f}$, denoted by δ. Fig 2 (left) shows the ideal δ field in a hypothetical alignment example. Analogously, the corresponding two rows of the Jacobian will be denoted by D. The component D_p quantifies up to first order the effect of parameter p.

In this continuous setting the gradient and Hessian of the Gauss-Newton update rule become:

$$J^T\mathbf{f} = \oint_{\mathbf{x}\in\partial T} J(\mathbf{x}) f(\mathbf{x}) dx \tag{1}$$

$$J^T J = \oint_{\mathbf{x}\in\partial T} J^T(\mathbf{x}) J(\mathbf{x}) dx \tag{2}$$

In terms of δ and D_p, and for a polygonal contour with segments S_k joining nodes k and $k+1$, the components of the gradient and Hessian can be expressed as:

$$\{\nabla C\}_p = \sum_{k=1}^{n} \int_{\mathbf{x}\in S_k} D_p(\mathbf{x}) \cdot \delta(\mathbf{x})\, dx \tag{3}$$

$$H_{pq} = \sum_{k=1}^{n} \int_{\mathbf{x}\in S_k} D_p(\mathbf{x}) \cdot D_q(\mathbf{x})\, dx \tag{4}$$

The δ field is useful to provide a geometric interpretation of the optimization process (Fig 2, left). The correction $\Delta\mathbf{p}$ is based on the accumulation along the whole contour of the scalar products $D_p \cdot \delta$. The locations in which they point to the same (opposite) direction support the fact that increasing (decreasing) this particular parameter the alignment error will be reduced. If they are nearly orthogonal the effect of p to improve alignment is negligible. The inverse Hessian is needed to coordinate possibly conflicting effects of different transformation parameters.

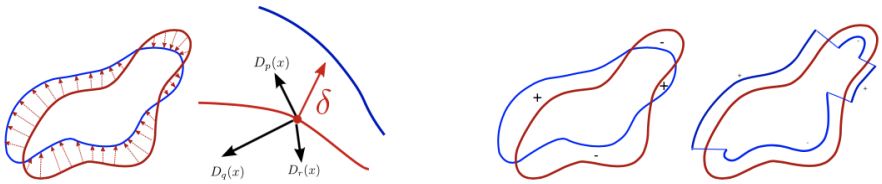

Fig. 2 Left: the vector field $\delta_{\mathbf{p}}(\mathbf{x})$ shows the local deformation required to improve alignment and the effects of the transformation which must be combined to match δ . Right:Signed XOR alignment error between a template T and a observed shape I, and the corresponding average residuals δ.

We use the areas of the mismatched regions, computed by the signed XOR clipping operation (Fig. 2, right), to provide information about the amount of local deformation required for alignment. Since we do not have landmarks or corresponding points the local alignment error δ can only be estimated as some kind of average distance between the contours in each mismatched region[2]. This can be easily ob-

[2] Active contours scan the normal to the contour in the image until they find and edge. In contrast, we obtain an average displacement in closed form just from the template, which becomes more precise in successive iterations.

tained as the area of that region divided by the length of the corresponding section of the contour. Fig. 3 (left) shows the δ field for an illustrative mismatch region represented by a polygonal approximation.

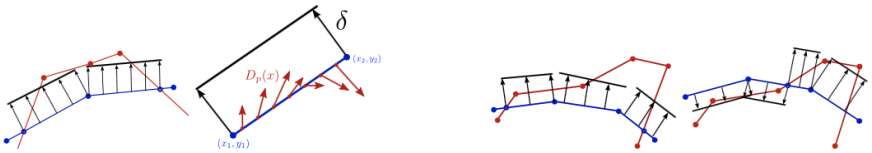

Fig. 3 Left: assignation of δ. Right: contribution to eq. (3) on the k-th segment. Right: Average error in successive steps.

This apparently crude estimation of the local error as average distance on the whole region is nevertheless extremely useful and easy to compute. Large mismatched regions with different contour distances usually take only one optimization step to be divided into more uniform regions in which the average estimation is more accurate.

Once the δ field is available from XOR polygon clipping, eqs. (3) and (4) reduce to simple integrals over piecewise linear sections with constant δ, that can be obtained in closed form in terms of the vertex coordinates (Fig. 3, right).

Consider the k-th segment δ_k from point (x_k, y_k) to (x_{k+1}, y_{k+1}) . The p-th element of the gradient is

$$\{\nabla C\}_p = \sum_k G_k \tag{5}$$

where the contribution of each segment can be expressed as

$$G_k = \int_{x_k}^{x_{k+1}} \delta_k D_p(x) = \delta_x X_k^p + \delta_y Y_k^p \tag{6}$$

in terms of the accumulated effect of the transformation:

$$X_k^p = \int_0^1 \frac{\partial x}{\partial p}(x_k(t), y_k(t))dt \tag{7}$$

$$Y_k^p = \int_0^1 \frac{\partial y}{\partial p}(x_k(t), y_k(t))dt \tag{8}$$

In the above expression $(x_k(t), y_k(t))$ is a parameterization of the segment from (x_k, y_k) to (x_{k+1}, y_{k+1}).

This approach requires very low computational effort compared to the 2D image processing operations required by the standard LK [8] approach. Since the global alignment area works without the need of point correspondences, we do not need a big number of vertexes in the polygonal approximation to the regions.

The initial state for the optimization is obtained from an affine invariant canonical frame obtained by whitening, which can also be computed in closed form from the contour vertexes. Rotation ambiguity can be eliminated by looking for the points in the whitened contour at extreme distances from the origin. For rigid templates the method must only estimate the non-affine component of the transformation.

3 Deformation modes

Rigid templates are unsatisfactory for many practical applications. On one hand, many shapes have different versions which cannot be modeled by affine or projective transformations (e.g., thickness or relative lengths of alphanumeric characters). There is a continuous set of possible shape variants that cannot be captured by a finite set of fixed prototypes. A more natural approach is to align a deformable template to the observed shape: from a single template we can extract both the image transformation parameters (with information about camera pose), and also the deformation parameters, which may be useful to identify the observed template version. On the other hand, deformable templates can be useful to model special observation circumstances such as curved surfaces and small occlusions or self-occlusions.

We will adopt a linear deformation model comprised by a base polygon and a set of variation modes described as vectors attached to each vertex (Fig. 4). This model is general enough to describe artificial markers with variable dimensions attached to curved surfaces, and can be easily incorporated to the previous alignment framework.

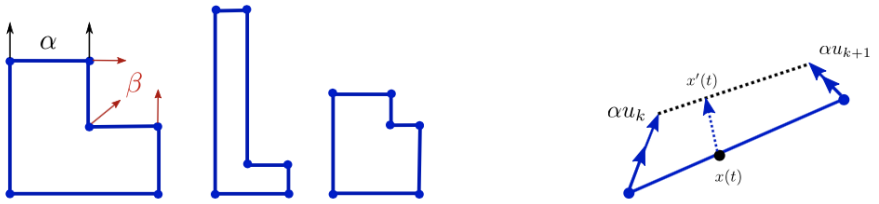

Fig. 4 Left: Linear deformation model expressed as a base shape (blue) and two deformation modes (black and red) with two example instances T (2,-1) and T (-1, 1) generated by this model. Right: Linear interpolation in a deformation mode.

The vertexes of the template are generated by a linear combination of the deformation parameters:

$$\mathbf{T}(\alpha) = \mathbf{T_0} + \alpha_1 \mathbf{u} + \alpha_2 \mathbf{v} + \ldots \tag{9}$$

The contour at a particular location parametrized by $t \in [0,1)$ along the k-segment is obtained by linear interpolation of the base figure and the deformation vectors:

$$\mathbf{x^k}_{\alpha_1,\alpha_2,\ldots}(t) = t(\mathbf{x_k} + \alpha_1\mathbf{u_k} + \alpha_2\mathbf{v_k} + \ldots) + \\ + (1-t)(\mathbf{x_{k+1}} + \alpha_1\mathbf{u_{k+1}} + \alpha_2\mathbf{v_{k+1}} + \ldots) \quad (10)$$

In order to incorporate the deformation parameters α into the framework developed in Sect. 2 we must only compute the integrals of eq. (7) for the gradient, and for the Hessian. Because of the linear nature of the deformation, the first ones are proportional to the average of the deformation vectors attached to the segment (of length l_k):

$$\begin{bmatrix} X_k^\alpha \\ Y_k^\alpha \end{bmatrix} = \frac{\mathbf{u^{(k)}} + \mathbf{u^{(k+1)}}2}{l}_k \quad (11)$$

There are now two kinds of parameters: p_j for the image transformation, and α_k for the deformation modes, so the integrals required by the Hessian are of three types. The products for $p_j\ p_k$ are computed as is explained in [1]. The products for $\alpha_i \sim (\mathbf{u^1}, \mathbf{u^2})$ and $\alpha_j \sim (\mathbf{v^1}, \mathbf{v^2})$, and the mixed products for $p \sim M(s,a,b,t,c,d)$ and $\alpha \sim (\mathbf{u}, \mathbf{v})$ can again be expressed in closed form in terms of the vertex coordinates and a new moment

$$J(w,z,n,m) \equiv \int_0^1 (tw + (1-t)z)x(t)^n y(t)^m \quad (12)$$

where $x(t)$ and $y(t)$ is a linear parametrization for $t \in [0,1)$ of the k-segment (from (x_k, y_k) to (x_{k+1}, y_{k+1})).

The linear deformation model is not a group (we cannot "remove" the estimated $\Delta\alpha$ from the observed image, we can only add it to the template), and therefore we cannot apply the more efficient inverse compositional LK variant. For computational convenience in our prototype we apply a mixed strategy, using inverse compositional update for the image warping parameters and forward additive update for the deformation modes (Fig. 5). The two sets of updates converge to the deformed shape actually observed, with the projective warped removed.

4 Experiments: Quadrotor experiments for accuracy validation

For the method validation we have designed two cylindrical bars with several patterns placed over them. These bars contain ARTags over both sides and another grid of points is placed in the middle. We assume that we have a precise 3D model of the objects. In our case, all necessary measurements are taken with a digital caliper (with precision of 0.01mm).

We propose two different configurations to validate the method. For the first configuration (Section IV-A), we show a realistic case in an outdoors scenario where there is a certain structure with two bars on it. The method extracts the pose of each

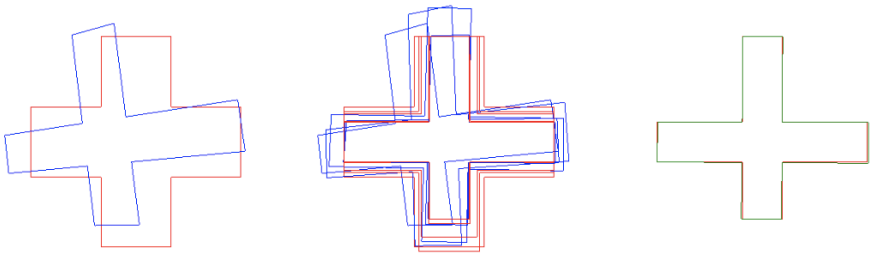

Fig. 5 Illustration of the mixed update alignment strategy. (a) Starting point with the observed contour (blue) and the template in neutral position (red). It has two deformation modes: thickness of the bars, and length of the lower one. (b) Additive forward updates for the deformation modes (red) and inverse compositional updates for the warping parameters (blue). (c) The matching result after 5 steps, with the following sequence of XOR alignment errors: (0.65,0.42,0.22,0.05,0.01).

bar using one or two markers per bar (depending on the visibility). In this case, we use a handheld camera to produce more challenging lightning conditions (not easily retrieved with the quadrotor). For the second configuration (Section IV-B), we will use a quadrotor with an attached camera to calculate the precision of the method and compare with other results from other methods. Our method will only work with the ARTags, not using the central grid at all. The grid will only be used for ground-truth calculation in Section IV-B.

In this section we evaluate the accuracy of the proposed method in a quadrotor with different scenes and bars configurations. We have developed an implementation in ROS of the method. For the purpose of accuracy evaluation we will show the design of these experimental setups and the calculation of a reliable ground-truth for further validation of the method. Finally, we will provide some error measurements with respect to the ground-truth as well as some images extracted from the method.

1) Experimental setup: For these experiments we will use a Pelican quadrotor with an attached camera of 752x480 pixels of resolution and 4mm of focal length. After different camera configurations this one has proven to be good enough for our experiments. The experimental setup consists of a flight area of approximately 3m where a big planar grid pattern (A3 size) altitude calculation as part of the ground-truth, and also for the camera calibration. Then, we place a prism of plastic of 30 and 60 degrees of slope, respectively (Fig. 6). Two bars are arranged forming different angles between them for each scene type.

For ground-truth calculation we use the middle grid pattern. We extract 25 2D-3D point correspondences by hand for each frame (we avoid unnecessary errors produced by automatic detection processes) and obtain the pose using EPnP [6] and Lu and Hager method for further refinement [7]. After that, we reproject the axis and other known 3D points of the bar model (not used for the pose calculation) to make sure that the result is correct. The method detects both ARTags and aligns the template with the deformations, obtaining another pose for each bar. Finally, we can

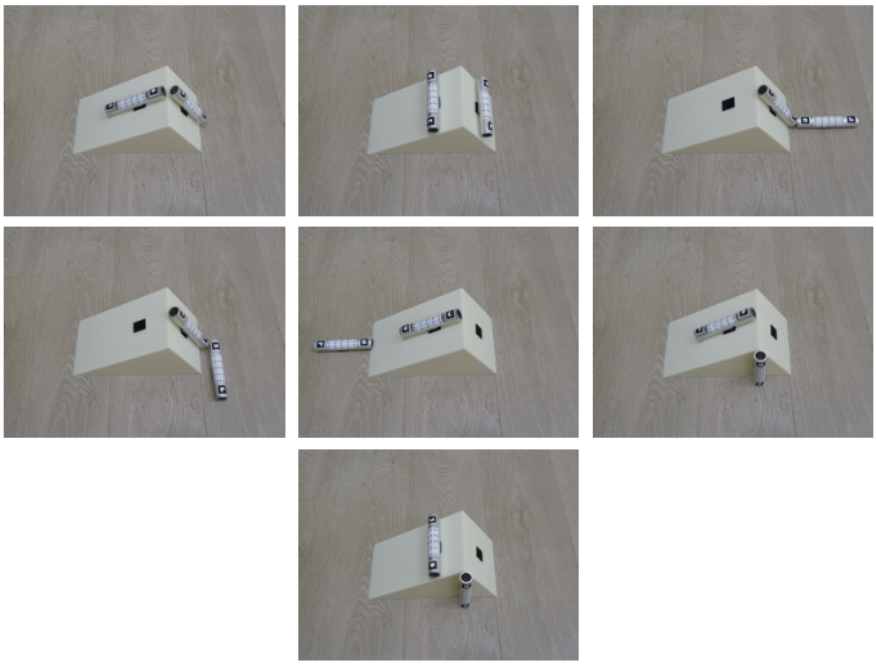

Fig. 6 Different bar configuration

evaluate the true error by just comparing with the ground-truth. This experiment is repeated for 7 different bar configurations, all of them shown in Fig. 6.

2) Results: The results can be summarized in the figure above Fig. 7). The ground-truth is correctly calculated as expected because we have used almost perfect measurements with nearly zero error. Also, the figure shows the quadrotor real trajectories in altitude and the average altitude for each experimental setup. The altitude data is very important for precision evaluation because it influences the marker occupancy in the image.

The zoomed-in image (top-right) shows the alignment error. The method is close to the ground-truth, even though the resolution is really low at this level of detail. Finally, we translate the quantitative results into Table I. We show absolute and relative errors for translation, because marker occupancy, camera resolution and precision are correlated.

5 Conclusions

In this chapter we have presented a relative localization method based on optimizing the alignment of deformable contours from texture-less images working from the

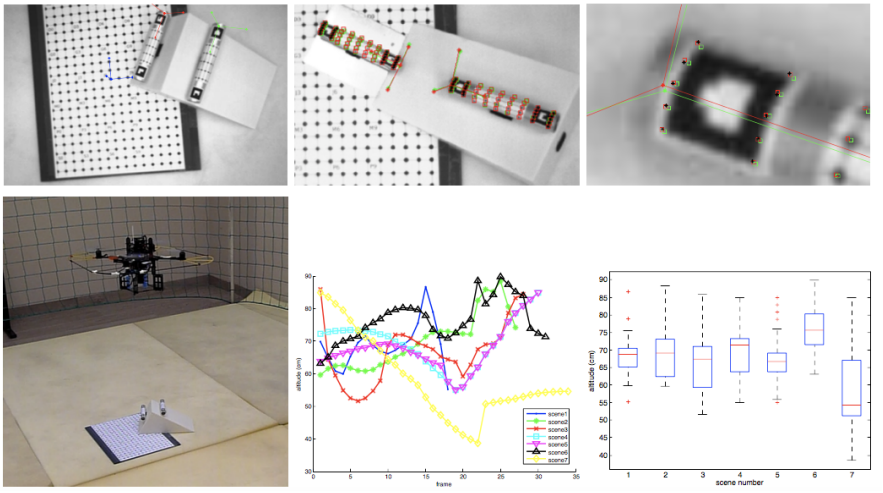

Fig. 7 Top left: Ground truth representation of the different coordinate systems. Top-middle: Comparison with ground-truth showing the reprojection of the 3D points with the pose calculated from the 3D to 2D correspondences. The image shows: ground-truth (green squares), proposed method results (red squares) and points obtained by the alignment (black stars). Top-right: Zoomed-in version of previous image, showing one side of a bar. Bottom-left: Quadrotor scene image taken from outside. Bottom-middle: Trajectories (altitude) of the quadrotor for the different scenes. Bottom-right: Average altitudes.

	ϵ_{abs} (mm)	ϵ_{rel}	Yaw	Pitch	Roll
μ	4.29	0.77%	4.94°	0.70°	0.99°
σ	2.21	0.38%	3.70°	0.44°	0.58°
ARToolkit	5-26	0.83-4.33%	-	-	-

Fig. 8 TABLE I: Average and standard deviation errors of the proposed method for the quadrotor experiment. ARToolkit errors were extracted from the benchmark in the website.

raw vertexes of the observed contour. The algorithm optimizes the alignment of the XOR area computed by means of computer graphics clipping techniques. To the best of our knowledge this geometric approach has not been studied before, even though it provides a very natural measure of alignment error without the need of correspondences. Our experiments show that the method provides very precise pose estimations in indoors and outdoors, showing very competitive results and proving itself as a low cost alternative to infrared motion capture systems. The experiments with our method yields an average error of less than 5mm in position at a distance of 0.7m, using a visual mark of 17mm x 17mm. The method can run in real-time and in a low cost hardware.

References

1. A. Amor-Martinez, A. Ruiz, F. Moreno-Noguer, and A. Sanfeliu. On-board real-time pose estimation for uavs using deformable visual contour registration. In *2014 IEEE ICRA*, pages 2595–2601, May 2014.
2. F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello. Rune-tag: A high accuracy fiducial marker with strong occlusion resilience. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 113–120, 2011.
3. G. Greiner and K. Hormann. Efficient clipping of arbitrary polygons. *ACM Transactions on Graphics (TOG)*, 17(2):71–83, 1998.
4. P. K. Jain. Homography estimation from planar contours. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 877–884, 2006.
5. H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *1999.(IWAR99) Proceedings in Augmented Reality*, page 8594, 1999.
6. V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155166, 2009.
7. C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610622, 2000.
8. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th IJCAI*, 1981.
9. J. Nemeth, C. Domokos, and Z. Kato. Recovering planar homographies between 2d shapes. In *Proccedings of ICCV*, pages 113–120, 2009.