2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

On-board Real-time Pose Estimation for UAVs using Deformable Visual Contour Registration

Adrian Amor-Martinez[1], Alberto Ruiz[2], Francesc Moreno-Noguer[1] and Alberto Sanfeliu[1]
[1]Institut de Robòtica i Informàtica Industrial, Barcelona, Spain
[2]Dept. Informática y Sistemas, University of Murcia, Spain
[1]{aamor,fmoreno,sanfeliu,}@iri.upc.edu, [2]aruiz@um.es

***Abstract*— We present a real time algorithm for estimating the pose of non-planar objects on which we have placed a visual marker. It is designed to overcome the limitations of small aerial robots, such as slow CPUs, low image resolution and geometric distortions produced by wide angle lenses or viewpoint changes. The method initially registers the shape of a known marker to the contours extracted in an image. For this purpose, and in contrast to state-of-the art, we do not seek to match textured patches or points of interest. Instead, we optimize a geometric alignment cost computed directly from raw polygonal representations of the observed regions using very simple and efficient clipping algorithms. Further speed is achieved by performing the optimization in the polygon representation space, avoiding the need of 2D image processing operations. Deformation modes are easily included in the optimization scheme, allowing an accurate registration of different markers attached to curved surfaces using a single deformable prototype. Once this initial registration is solved, the object pose is retrieved using a standard PnP approach.**

As a result, the method achieves accurate object pose estimation in real-time, which is very important for interactive UAV tasks, for example for short distance surveillance or bar assembly. We present experiments where our method yields, at about 30Hz, an average error of less than 5mm in estimating the position of a 19x19mm marker placed at 0.7m of the camera.

I. Introduction

Image registration is at the core of many tasks in robotics, including object detection and manipulation, pose estimation, human-robot interaction or inter-robot interaction. In the context of UAV outdoor navigation, the feature-based image registration problem is hard to deal with because of light changes, images are distorted due to the aerial robot motion and vibrations, cameras usually have low resolution, lens small defects and the computational capabilities are frequently limited. Therefore, we need robust and simple feature detection or segmentation methods that can handle these issues. Moreover, when one seeks to obtain accurate pose estimations of textured or non-textured objects, we need to select what kind of features and geometry are required. In our case, we have selected artificial visual markers, like ARToolkit [1], to obtain high accuracy in object pose estimation. This is especially necessary in the tasks we describe in this paper of supervision and manipulation of non-textured bars using UAVs.

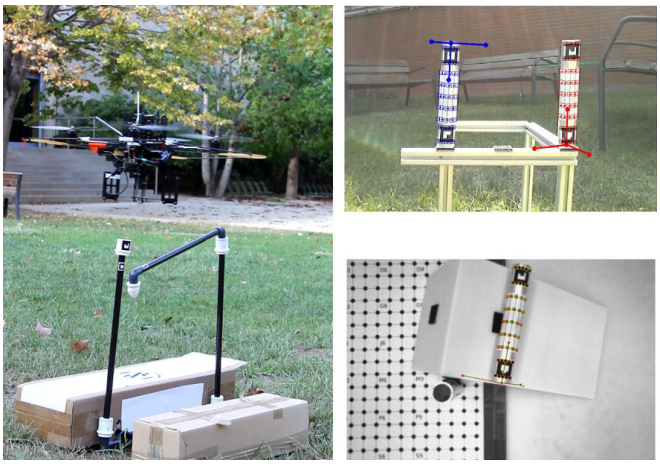


Fig. 1: Left: Case scenario we consider in this paper of a quadrotor under a supervision task. Right: Images of the bars acquired with the onboard cameras. Our goal is to recover the pose of the bar from the squared markers at the opposite sides of the bars. This kind of markers can be easily deployed 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.

Traditional registration methods like Lucas-Kanade (LK) [2], [3] are very powerful for textured images. However, when working on essentially binary images the optimization steps are only qualitative, and improvements like pyramids only help to solve for image displacements.

ARToolkit [1] is commonly used for augmented reality purposes, and requires corner extraction and rectification before recognizing the artificial markers. Several improvements have been proposed [4] to increase robustness to light changes, partial occlusion and inter-marker detection. We can find more recent work on fiducial markers in [5], that proposes an approach that produces very accurate results and shows robustness to strong occlusions thanks to the large number of dots and the redundancy included in the marker. The method uses coplanar circle rectification [6], but is constrained to planar surfaces to work properly.

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness under project TaskCoop DPI2010-17112, by the ERA-Net Chistera project ViSen PCIN-2013-047 and by the EU project ARCAS FP7-ICT-2011-28761. A. Ruiz is supported by FEDER funds under grant TIN2012-38341-C04-03.

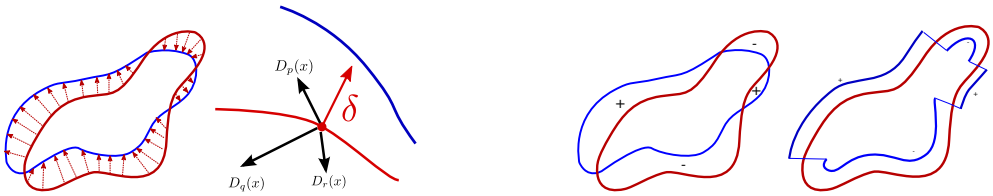


Fig. 2: Left: the vector field $\delta_{\boldsymbol{p}}(\boldsymbol{x})$ shows the local deformation required to improve alignment and the effects of the transformation which must be combined to match $\boldsymbol{\delta}$. Right: Signed XOR alignment error between a template T and an observed shape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

In contrast to making a particular design for each purpose, we propose a general contour alignment approach that meets the same requirements as fiducial markers but with extra precision and an extension to deformable contours.

The problem of general contour alignment has been widely studied in previous works. Active contours [7] use splines and observation vectors based on locating edges along the normal of the contour. Alignment is based on energy minimization [8] or dynamic programming [9]. Other approaches like Gradient Vector Flow [10] and its variants [11] perform better when the initialization is not as good. More recent approaches in contour alignment work on different contour representations. In [12] a polygon triangulation representation is used for the optimization, whereas [13] uses a different scheme based on hierarchical point locations. Besides being computationally expensive, these methods can not achieve the goal of providing reliable correspondences because they are image-based. Also, they require good gradient information, not available from binary images. Other methods for homography estimation from general planar contours have been proposed [14], [15], [16], [17], but they typically require complex optimization schemes and cannot be easily extended to deformable shapes.

Also related with our work, are the point alignment methods that seek to align two sets of points, either 2D-2D or 2D-3D. The well known Iterative-Closest-Point algorithm [18] can be used to register this transformation but is constrained to rigid deformations. Non-rigid warps have been addressed by recent approaches [19], [20], [21], but in all of them, textured surfaces are needed. This is not required in our contour-based approach.

More specifically, in this paper we have developed an efficient registration method for contours which is based on a Gauss-Newton optimization of a natural geometric alignment cost based on polygonal XOR clipping. The method is based on the whole image, does not need to search for correspondences and is 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 onboard solution for small object pose estimation (Fig. 1). In addition, our approach has the benefit over motion capture systems because they are not really appropriate for outdoors, where the environment is less controllable in most situations.

The rest of the paper is organized as follows. In Section 2 we describe the formulation for contour-based alignment. The method is extended to deformation modes in Section 3. In Section 4 we present an experimental validation of the approach for different real scene configurations using a quadrotor with an attached camera. The paper closes with some concluding remarks and future directions of this work.

II. 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 [22], contour extraction, and polygon reduction [23].

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 an XOR (symmetric difference) clipping algorithm working on region boundaries [24], and their areas can be easily computed just from the contour nodes.

Given a transformation model $\boldsymbol{x}' = \mathtt{W}_{\boldsymbol{p}}(\boldsymbol{x})$, the contour registration problem will be formulated as finding the parameters $\boldsymbol{p}$ that minimize the error area XOR(O, $\mathtt{W}_{\boldsymbol{p}}$(T)) for the observed contour O and template T. This can be solved using Gauss-Newton's iterative optimization: Given a residual vector $\boldsymbol{f}$ with Jacobian $\mathtt{J} = \partial \boldsymbol{f}/\partial \boldsymbol{p}$, the squared error $C = 1/2 \boldsymbol{f}^{\mathsf{T}} \boldsymbol{f}$ can be reduced by using the update rule $\Delta \boldsymbol{p} = -\mathtt{H}^{-1} \nabla C$, where $\nabla C = \mathtt{J}^{\mathsf{T}} \boldsymbol{f}$ and the Hessian is approximated by $\mathtt{H} = \mathtt{J}^{\mathsf{T}} \mathtt{J}$.

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 $\boldsymbol{f}$, denoted by $\boldsymbol{\delta}$. Fig 2 (left) shows the ideal $\boldsymbol{\delta}$ 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.

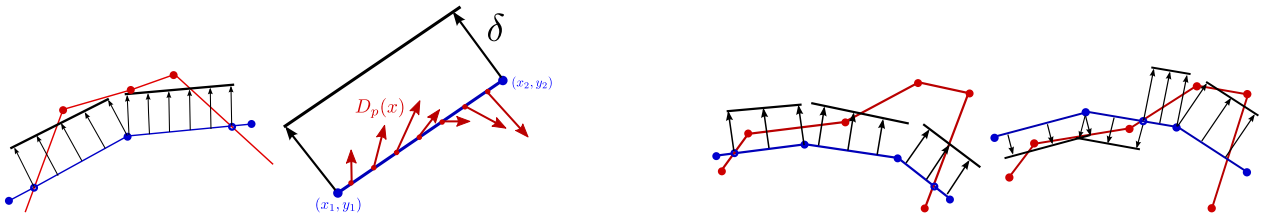


Fig. 3: Left: assignation of δ and contribution to eq. (3) on the k-th segment. Right: Average error in successive steps.

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

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

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

In terms of $\boldsymbol{\delta}$ 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:

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

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

The δ field provides a geometric interpretation of the optimization process (Fig 2, left). The correction $\Delta\boldsymbol{p}$ is based on the accumulation along the whole contour of the scalar products $D_p \cdot \boldsymbol{\delta}$. The locations in which they point to the same (opposite) direction support the fact that increasing (decreasing) this particular parameter the aligment 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.

We use the areas of the mismatched regions, computed by the signed XOR clipping operation (Fig. 2, right), to estimate the amount of local deformation required for alignment. Since we do not have landmarks or corresponding points the local alignment error δ is estimated at the average distance in perpendicular direction between the contours in each mismatched region[1]. This can be easily obtained 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.

This apparently crude estimation of the local error as average distance on the whole region is still very useful and easy to compute just from the template nodes. 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 (Fig. 3, right).

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}).

It can also be easily checked that the contribution of segment k to the element H_{pq} of the Hessian (approximated by $\mathtt{J}^\mathsf{T}\mathtt{J}$) is the following (the components of the piecewise constant template gradient are denoted by $\nabla T = (g_x, g_y)$):

$$H_{pq}^k = g_x^2 \int_k \frac{\partial x}{\partial p}\frac{\partial x}{\partial q} + g_x g_y \int_k \frac{\partial x}{\partial p}\frac{\partial y}{\partial q} + \\ + g_y g_x \int_k \frac{\partial y}{\partial p}\frac{\partial x}{\partial q} + g_y^2 \int_k \frac{\partial y}{\partial p}\frac{\partial y}{\partial q} \tag{9}$$

We will express the derivatives $D_p = (\frac{\partial x}{\partial p}, \frac{\partial y}{\partial p})^\mathsf{T}$ of the transformation W at $\boldsymbol{p} = \mathbf{0}$ in terms of the coefficients s and t and the exponents a, b, c, and d of a simple monomial as follows:

$$D_p = \begin{bmatrix} s x^a y^b \\ t x^c y^d \end{bmatrix} \equiv M(s, a, b, t, c, d) \tag{10}$$

This simple expression is general enough to accommodate all image transformations of interest, from simple displacements to projective warping. The derivatives are taken at the origin, where W is the identity transformation, which is

[1]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.

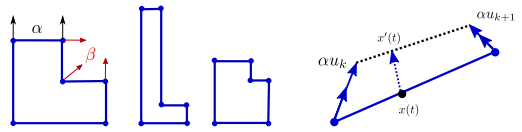


Fig. 4: Left: Linear deformation model expressed as a base shape (blue) and two deformation modes (black and red). Right: Linear interpolation in a deformation mode.

suitable for the inverse compositional update variant. For example, a rotation is $M(-1,0,1,1,1,0)$, the $h_{3,1}$ coefficient of a homography is $M(-1,2,0,-1,1,1)$, and so on.

The products required in eq. (9) for the Hessian reduce to the same monomial structure

$$s_1 x^{a_1} y^{b_1} s_2 x^{a_2} y^{b_2} = (s_1 s_2) x^{a_1+a_2} y^{b_1+b_2} \tag{11}$$

so we only need a closed form expression for the moment

$$I_k^{n,m} = \int_{(x_k,y_k)}^{(x_{k+1},y_{k+1})} x^n y^m \tag{12}$$

which can be easily obtained for the required exponents n and m using any computer algebra system[2].

Using this, the contribution to the gradient (6) of parameter $p \sim M(s,a,b,t,c,d)$ becomes

$$G_k = \delta_x s I_k^{a,b} + \delta_y t I_k^{c,d} \tag{13}$$

and the Hessian element for $p \sim M(s_1,a_1,b_1,t_1,c_1,d_1)$ and $q \sim M(s_2,a_2,b_2,t_2,c_2,d_2)$ can be expressed as:

$$H_{pq}^k = g_x^2 s_1 s_2 I_k^{a_1+a_2,b_1+b_2} + g_x g_y s_1 t_2 I_k^{a_1+c_2,b_1+d_2} + \dots \tag{14}$$

This approach requires very low computational effort compared to the 2D image processing operations required by the standard registration approaches. Since the global alignment area works without 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 nonaffine component of the transformation.

III. 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 comprising a base polygon and a set of variation modes described as vectors attached to each vertex (Fig. 4, left). 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.

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

$$\boldsymbol{T}(\boldsymbol{\alpha}) = \boldsymbol{T}_0 + \alpha_1 \boldsymbol{u} + \alpha_2 \boldsymbol{v} + \dots \tag{15}$$

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 (Fig. 4, right):

$$\begin{aligned} \boldsymbol{x}^k_{\alpha_1,\alpha_2,\dots}(t) = t(\boldsymbol{x}_k + \alpha_1 \boldsymbol{u}_k + \alpha_2 \boldsymbol{v}_k + \dots) + \\ + (1-t)(\boldsymbol{x}_{k+1} + \alpha_1 \boldsymbol{u}_{k+1} + \alpha_2 \boldsymbol{v}_{k+1} + \dots) \end{aligned} \tag{16}$$

In order to incorporate the deformation parameters $\boldsymbol{\alpha}$ into the framework developed in Sect. II we must only compute the integrals of eq. (7) for the gradient, and (9) 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):

$$\begin{bmatrix} X_k^\alpha \\ Y_k^\alpha \end{bmatrix} = \frac{\boldsymbol{u}^{(k)} + \boldsymbol{u}^{(k+1)}}{2} l_k \tag{17}$$

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 before using (11) and (12). The products for $\alpha_i \sim (\boldsymbol{u}^1, \boldsymbol{u}^2)$ and $\alpha_j \sim (\boldsymbol{v}^1, \boldsymbol{v}^2)$, and the mixed products for $p \sim M(s,a,b,t,c,d)$ and $\alpha \sim (\boldsymbol{u}, \boldsymbol{v})$ can again be expressed in closed form in terms of the vertex coordinates and a new moment

$$J(w,z,n,m) \equiv \int_0^1 (tw + (1-t)z) x(t)^n y(t)^m \tag{18}$$

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})). Explicit expressions for H_{ij}^k and $J(w,z,n,m)$ for typical exponents are included as supplementary material.

The linear deformation model is not a group (we cannot "remove" the estimated $\Delta\boldsymbol{\alpha}$ from the observed image, we can only add it to the template), and therefore we cannot apply the more efficient inverse compositional optimization variant.

[2]There is a general expression for $I_k^{n,m}$ in terms of a hypergeometric function but in practice it is much faster to use special solutions as in $I_k^{2,0} = (x_1^2 + x_2^2 + x_1 x_2)/3$, and so on.

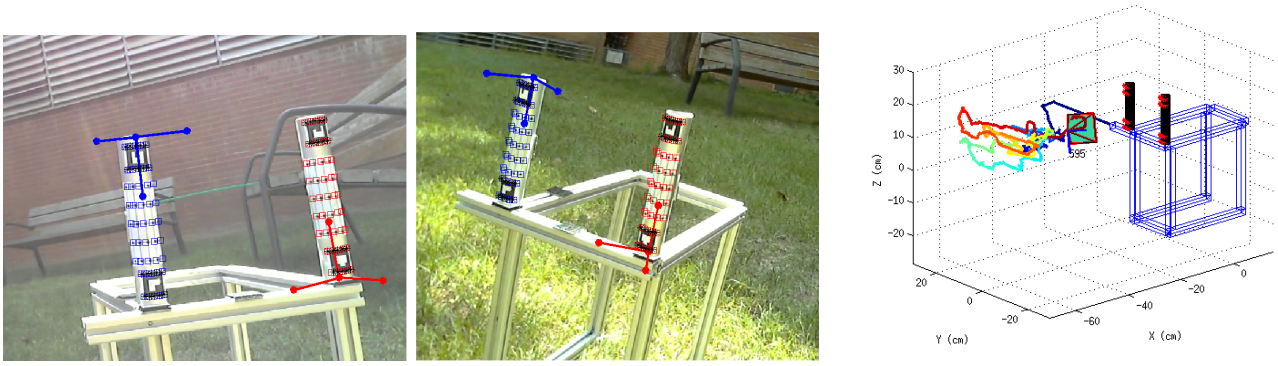


Fig. 5: Results of the algorithm for the handheld camera scenario. Left and center: different frames with projection of 3D points and bar coordinate system. Right: Camera representation and trajectory with respect to the 3D model.

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. 6). The two sets of updates converge to the deformed shape actually observed, with the projective warped removed.

As an example, Fig. 7 shows how we can use deformation modes for a squared contour placed parallel to a cylinder cross axis. In this case, the contour deformation only occurs on the left and right sides of the square. This model will be used for the experiments in the following section.

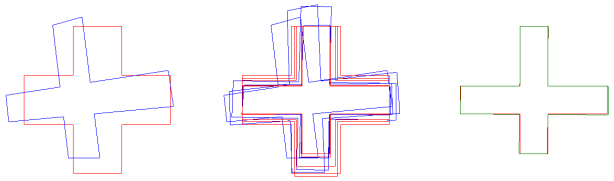
IV. Experiments

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 bar using one or two markers per bar (depending on the visibility). In this case, we use a handheld camera to produce more challenging illumination 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 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.



(a) (b) (c)

Fig. 6: Illustration of the mixed update alignment strategy with two deformation modes. (a) Observed contour (blue) and template (red). (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).

A. Handheld camera real experiments

We will use a webcam with a down-scaled resolution of 800x448 pixels. The main results are summarized in Fig. 5. Firstly, we run our contour alignment method to detect the internal shape so the method can identify each one of the markers (any other method is valid in this case, because for the identification there is no need for extra precision). The contour alignment method provides a camera pose for each one of the bars, using 10 points for each marker. Secondly, we project the 3D points of the central grid, the system coordinates and the marker itself. Finally, we can draw the trajectory of the camera with respect to the 3D model of the structure. As we see, the precision is very good since the squares are very close to the dots in the central grid.

B. Quadrotor experiments for accuracy validation

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. 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

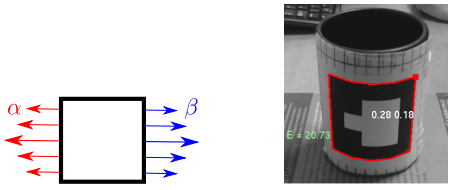


Fig. 7: Deformable model for a square over a cylinder

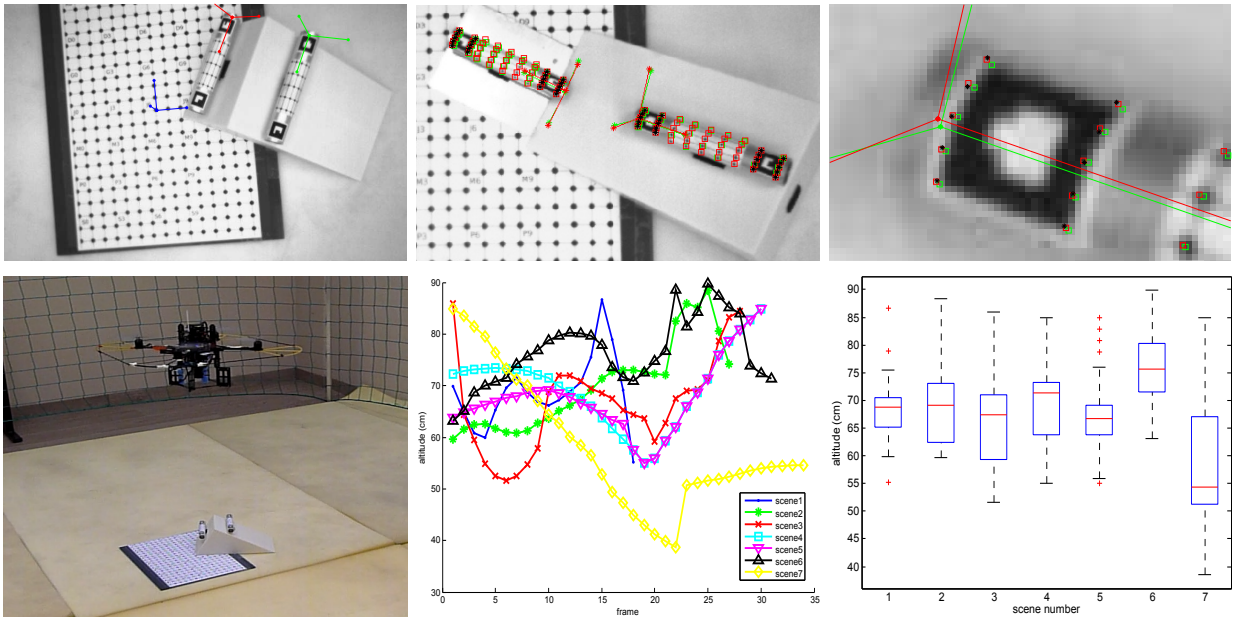


Fig. 8: Top left: Ground truth representation of the different coordinate systems. Top-middle: Comparison with ground-truth showing the reprojection of the 3D points with the pose calculated from the 3D to 2D correspondences. The image shows: ground-truth (green squares), proposed method results (red 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.

provide 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 (See Fig. 8-bottom-left). After different camera configurations this one has proven to be good enough for our experiments[3].

The experimental setup consists of a flight area of approximately $3m^3$ where a big planar grid pattern (A3 size) is placed on the floor. This pattern will be used for a precise 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. 9). 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 detection processes) and obtain the pose using EPnP[25] and Lu & Hager method for further refinement[26]. 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 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. 9.

2) Results: The results can be summarized in the figure above (Fig. 8). 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. The relative error is calculated as:

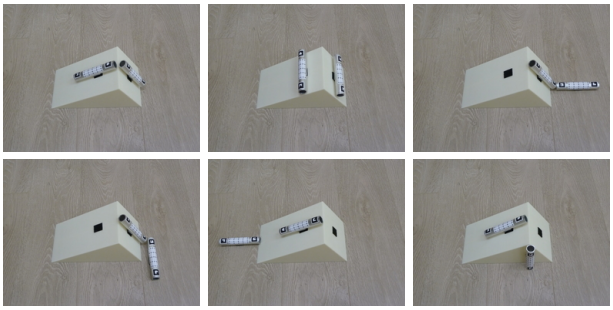


Fig. 9: Different bar configurations

[3]In UAVs we have additional problems with cameras because a tiny field of view can lead to more detail of the region of interest but is more unstable as this region of interest can be easily lost for the camera. We finally chose a small focal length at the cost of having less marker occupancy in the images to ensure that the marker is visible most of the time.

	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%	-	-	-

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.

	Proposed method	Infrared motion capture
Frequency	30 Hz	80-300 Hz
Precision	4.29 mm	∼0.5mm
Number of cameras	1	15-25
Suitable for outdoors	Yes	No

TABLE II: Comparison with global motion estimation systems. Precision for motion capture depends on the working area size.

$$\epsilon_{\text{rel}}^{(i)} = \frac{\epsilon_{\text{abs}}^{(i)}}{\|T_{\text{true}}^{(i)}\|_2}$$

Note that we can achieve a 0.77% of relative error. This means that, at a distance of 10cm, we have 0.8mm of error, which can satisfy UAV manipulation requirements where the precision is very important. Other methods like ARToolkit can not achieve this precision in these imaging conditions.

This method can be used as a low cost alternative to other high precision systems like infrared-based motion capture systems, as shown in Table II.

V. Conclusions

In this paper we have presented a new method for object pose estimation from an UAV using visual landmarks that do the computation in real time, it is based on onboard vision and obtains high pose precision. Moreover the method can handle deformable contour aligment from textureless images working from the raw vertexes of the observed contour. In contrast with the standard techniques based on corresponding image features, our method considers the true alignment error of the contours. The algorithm optimizes the alignment of the XOR area computed by means of computer graphics clipping techniques. This method allows for real-time applications on low cost hardware. We can work over a reduced set of vertexes and compute in closed form all the necessary magnitudes for Gauss-Newton optimization. Additionally, we can estimate deformations over strongly projective views.

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 explicit 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. This is very useful for supervision and assembly tasks in UAVs because we can achieve very high precision at close distances. Also, it is very appropriate for outdoors where the illumination conditions are continuously changing and there is extra noise because of the flying movement.

Future work includes registration of 3D contours and low level optimizations of the implementation.

References

[1] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Int. Workshop on Augmented Reality*, 1999.

[2] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981.

[3] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. Journal of Computer Vision*, vol. 56(3), 2004.

[4] M. Fiala, "Designing highly reliable fiducial markers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32(7), 2010.

[5] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in *Conf. on Computer Vision and Pattern Recognition*, 2011.

[6] Q. Chen, H. Wu, and T. Wada, "Camera calibration with two arbitrary coplanar circles," *European Conf. on Computer Vision*, 2004.

[7] A. Blake and M. Isard, *Active Contours*. Springer, 1998.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. Journal of Computer Vision*, vol. 1(4), 1988.

[9] D. Geiger, A. Gupta, L. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17(3), 1995.

[10] C. Xu and J. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. on Image Processing*, vol. 7(3), 1998.

[11] F. Moreno-Noguer, A. Sanfeliu, and D. Samaras, "Integration of deformable contours and a multiple hypotheses fisher color model for robust tracking in varying illuminant environments," *Image and Vision Computing*, vol. 25(3), 2007.

[12] P. Felzenszwalb, "Representation and detection of deformable shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27(2), 2005.

[13] P. Felzenszwalb and J. Schwartz, "Hierarchical matching of deformable shapes," in *Conf. on Computer Vision and Pattern Recognition*, 2007.

[14] J. Nemeth, C. Domokos, and Z. Kato, "Recovering planar homographies between 2d shapes," in *Int. Conf. on Computer Vision*, 2009.

[15] A. Ruiz, P. de Teruel, and L. Fernández, "Robust homography estimation from planar contours based on convexity," in *European Conf. on Computer Vision*, 2006.

[16] P. Jain, "Homography estimation from planar contours," in *Symposium on 3D Data Processing, Visualization, and Transmission*, 2006.

[17] S. Zhai Fu J. Zhai and Z. Jing, "Homography estimation from planar contours in image sequence," *Opt. Eng. 49*, 2010.

[18] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. Journal of Computer Vision*, vol. 13(2), 1994.

[19] E. Serradell, A. Romero, R. Leta, C. Gatta, and F. Moreno-Noguer, "Simultaneous correspondence and non-rigid 3d reconstruction of the coronary tree from single x-ray images," in *Int. Conference on Computer Vision*, 2011.

[20] F. Moreno-Noguer, J. M. Porta, and P. Fua, "Exploring ambiguities for monocular non-rigid shape estimation," in *European Conf. on Computer Vision*, 2010.

[21] J. Sanchez, J. Ostlund, P. Fua, and F. Moreno-Noguer, "Simultaneous pose, correspondence and non-rigid shape," in *Conf. on Computer Vision and Pattern Recognition*, 2010.

[22] A. Savakis, "Adaptive document image thresholding using foreground and background clustering," in *Int. Conf. on Image Processing*, 1998.

[23] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Canadian Cartographer*, 1973.

[24] G. Greiner and K. Hormann, "Efficient clipping of arbitrary polygons," *ACM Transactions on Graphics*, vol. 17(2), 1998.

[25] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative o(n) solution to the pnp problem," in *Int. Conf. on Computer Vision*, 2007.

[26] C. Lu, G. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22(6), 2000.