Geo-referencing for UAV Navigation using Environmental Classification

Fredrik Lindsten, Jonas Callmer, Henrik Ohlsson, David Törnqvist,
Thomas B. Schön and Fredrik Gustafsson
Division of Automatic Control, Department of Electrical Engineering
Linköping University, Sweden
{lindsten, callmer, ohlsson, tornqvist, schon, fredrik}@isy.liu.se

***Abstract*— A UAV navigation system relying on GPS is vulnerable to signal failure, making a drift free backup system necessary. We introduce a vision based geo-referencing system that uses pre-existing maps to reduce the long term drift. The system classifies an image according to its environmental content and thereafter matches it to an environmentally classified map over the operational area. This map matching provides a measurement of the absolute location of the UAV, that can easily be incorporated into a sensor fusion framework. Experiments show that the geo-referencing system reduces the long term drift in UAV navigation, enhancing the ability of the UAV to navigate accurately over large areas without the use of GPS.**

I. Introduction

Navigation of commercial UAVs is today depending on Global Navigation Satellite Systems, e.g. GPS. However, to solely rely on GPS is associated with a risk. When operating close to obstacles, reflections can make the GPS signal unreliable and it is also easy to jam the GPS making it vulnerable to malicious attacks. The navigation system thus requires an additional position estimator, allowing the UAV to keep operating even after GPS failure.

A sensory setup using an inertial measurement unit (IMU) together with vision from an on-board camera has been shown to enable accurate pose estimates through the process of visual odometry (VO) fused with an IMU [9]. However, without any absolute position reference the estimated position of the UAV will always suffer from a drift. The drift problem can be addressed using Simultaneous Localization And Mapping (SLAM) [1, 3] which relies on revisiting familiar areas to obtain so called loop closures. This means that the UAV needs to map its operational environment while operating in closed loops to minimize drift. This is of course a major drawback with SLAM for applications in which it is not natural to operate in closed loops.

We propose to use existing, preclassified maps of the operational environment for absolute position referencing, see Fig. 1. Using existing maps as reference instead of creating a new map online results in more accurate navigation and lets the UAV exploit what we already know. In this work we explore a vision based approach where images from the on-board camera are matched with the map, requiring no additional sensors apart from those used in VO. A similar idea was proposed in [2] where Normalized Cross Correlation (NCC) is used to correlate the on-board image with the reference map. We shall come back to this later. Also [8] address the problem, where reference image matching using the Hausdorff measure was explored. That work is mainly focused on the image processing properties and it is not incorporated into a probabilistic sensor fusion framework.

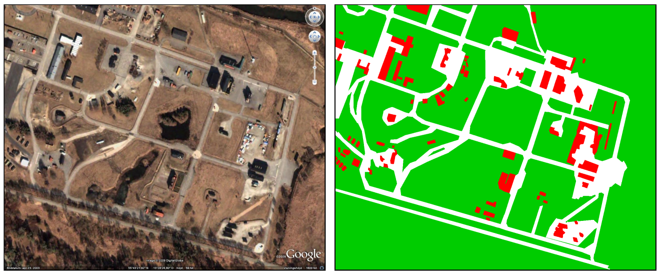

Fig. 1. Map over the operational environment obtained from Google Earth$^{\text{TM}}$ (left) and a manually classified reference map with grass, asphalt and houses as prespecified classes (right).

The idea behind geo-referencing is to provide a measurement equation, relating the on-board image I_t to the absolute position of the UAV,

$$y(I_t) = h(x_t) + e_t, \tag{1}$$

where $y(I_t)$ is some measurement derived from the image, x_t denotes the state and e_t denotes the measurement noise. $h(x_t)$ is a measurement model available as a look-up table based on the reference map. It is clear that I_t will depend on the full pose of the vehicle in 6 degrees of freedom and in the general case this should be the case for $h(x_t)$ as well. However, it is not feasible to use a 6D look-up table, which means that some approximations and/or simplifications are needed.

Since the reference map is available as a 2D image as shown in Fig. 1, we seek a measurement model h which only depends on the pixel coordinates in the map, $[u, v]$. The pose is related to these coordinates due to the fact that for a given pose we can project the on-board image onto the reference map, and obtain the coordinates $[u, v]$ corresponding to the centre of the projected image I_t. By doing so we enforce our measurement model, which now takes the form $h(u(x_t), v(x_t))$, to yield the same output for all vehicle poses resulting in the same pixel coordinates. Clearly, this must also be the case for the measurement $y(I_t)$, which means that the on-board image must be matched with the reference map in a way that only depends on $[u, v]$.

There are basically two ways to achieve this. The first is to allow the measurement to depend on the vehicle pose as well, i.e. $y(I_t, x_t)$. The problem with this approach is that we do not know the true pose, and when computing the measurement online we have to use an estimate. This approach is investigated in [2], where I_t is rotated and scaled using the current pose estimate to match the reference map. NCC is thereafter used to perform the matching in 2D.

The problem is that this method can result in instability if the pose estimate starts to drift, as shown in [2]. The second alternative is to make the matching invariant to rotation and/or scale. This is in itself not an approximation and does not suffer from instability issues. The price for using invariant matching is instead that some information is discarded and the geo-referencing becomes less informative.

In our proposed approach, the matching is made invariant to rotation and the scale is taken from a point estimate. The reason for this is that the measurement is believed to vary smoothly with respect to the scale, and the matching will thus be less sensitive to approximation errors in scale than orientation. Consider for instance the case where the UAV is flying along a road. Even a small error in rotation can then lead to a poor match when the on-board images are compared with the map. A small error in scale will not affect the matching as much. In our experiments we have small attitude angles and the scale will thus only depend on the altitude z_t. The idea can however easily be extended to the case where also point estimates of the attitude angles are used to compute the measurement.

II. Geo-referencing

Our geo-referencing framework uses environmental classification and rotation invariant template matching. The main motivation for using environmental classification and classified maps instead of aerial photos and point feature matching, is to gain robustness in the geo-referencing in the sense that it is insensitive to for instance daylight and even seasonal variations. Additional motives for performing the classification could be to assist in decision making, e.g. a UAV searching for a landing site must be able to distinguish between houses, forest, flat ground etc.

The basic procedure is as follows. I_t is first segmented and classified into houses, roads, grass etc. The classifier provides class probabilities for all segments. To describe the content of I_t in a rotation invariant way a class histogram $y(I_t)$ is computed from a circular region in the image. The histogram represents the proportions of the different classes in the circular region, which will be unaffected by any rotation of the image. A noise distribution for e_t, representing the uncertainty in the classification, is also derived. A flow chart of the procedure is provided in Fig. 2.

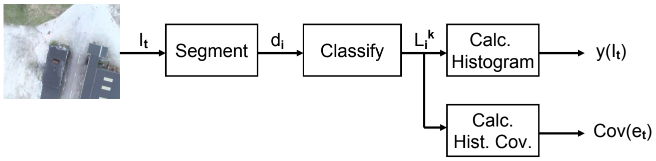

Fig. 2. Flow chart of the process of creating the measurement $y(I_t)$.

To enhance the template matching performance, the image is divided into N circular regions instead of just one (see Fig. 3), each for which a class histogram is computed. The same procedure applies to the reference map, for which the N histograms are precomputed offline at each pixel. The radii of the regions in I_t depend on the altitude estimate $\hat{z}_t$ so that their scales match the regions in the reference map.

We now have the measurement equations

$$y_n(I_t, \hat{z}_t) \approx y_n(I_t, z_t) = h_n(u(x_t), v(x_t)) + e_{n,t}, \qquad (2)$$

for $n = 1, \ldots, N$, where y_n and h_n are the class histograms for the n:th circular regions in I_t and in the reference map at position $[u, v]$, respectively. At this point one could find it strange that we have assumed additive noise e_t in a model dealing with class histograms. However, as we shall see in Sec. II-B this choice is well motivated. We shall also see that the main challenge in this approach is to find a proper distribution for e_t which reflects the uncertainties induced by the classification procedure.

A. Environmental Classification

The environmental classification of an image is initiated by the segmentation of the image into uniform regions called superpixels, using an off-the-shelf graph-based image segmentation algorithm [4]. We then seek the class probabilities for each superpixel

$$p_i(C^k|d_i) = P(\text{“superpixel } i\text{”} = C^k), \qquad (3)$$

for a set of prespecified classes C^k, $k = 1, \ldots, K$, where d_i is a descriptor of superpixel i. The classes are chosen with respect to the reference map, so that classes present in the map also become “available” to the classifier. This also means that we only consider classes that are believed to be more or less stationary, such as houses and roads. Using objects that are believed to be non-stationary, e.g. cars, will not work since these objects will most likely not be present in the reference map. The classes used in this work are grass, asphalt and house.

Each descriptor d_i is here taken as a 39 dimensional vector representing a superpixel. The color information contained in d_i is the RGB mean and variance (3x2 dim) and a histogram representation of the RGB content (3x8 dim) in the superpixel. Texture is incorporated using Gabor filtering with two scales and two directions. The mean and variance of each Gabor filtering is included in the descriptor (2x2x2 dim) and finally also the size of the superpixel (1 dim).

For classification, a neural network with 20 hidden units is trained to classify a descriptor d_i as one of the K classes. The network is trained with 594 manually labeled superpixels from 50 frames, not used in the validation or experiment data sets. When classifying a new descriptor, the output from the neural network is

$$L_i^k \in [0, 1], \; k = 1, \ldots, K, \qquad (4)$$

where $L_i^k = 1$ for some k implies a very certain classification. To be able to interpret the output as probabilities the L_i^k:s are normalized to sum to one, yielding

$$p_i^k \triangleq p_i(C^k|d_i) = L_i^k / \textstyle\sum_{l=1}^K L_i^l. \qquad (5)$$

Our classifier was validated using 166 superpixels, which resulted in a classification accuracy of 95%. In Fig. 3 the segmentation and classification of an image can be studied, where the class assigned to each superpixel is the one with the highest probability p_i^k. It is important to emphasize that neither the choice of descriptor nor classifier is central to the geo-referencing system presented here. The framework

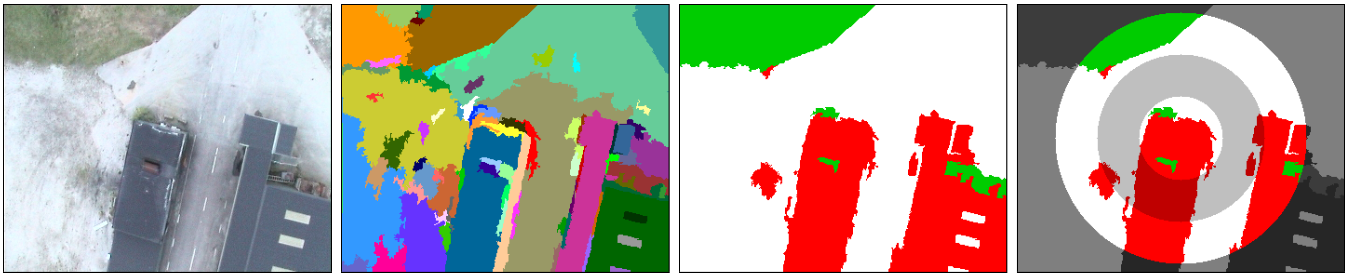

Fig. 3. Image from on-board camera (left), extracted superpixels (middle-left), superpixels classified as grass, asphalt or house (middle-right) and three circular regions used for computing the class histograms (right).

can be used with any other probabilistic classifier without modification, see for example [5, 6].

B. Probabilistic Template Matching

We now turn to the problem of finding a class histogram $y(I_t, \hat{z}_t)$ and a noise distribution for e_t reflecting the uncertainty in the classification. To do this we associate a stochastic variable $X_{i,t}$ to each superpixel representing its class, such that $X_{i,t}$ takes on class C^k with probability p_i^k. Here $p_i = \begin{pmatrix} p_i^1 & \dots & p_i^K \end{pmatrix}^T$ are the class probabilities given by the classifier. The classes are coded using a 1-of-K coding scheme, i.e. C^k is a binary vector, where the k:th element equals one and all other elements equal zero,

$$C^k = \underbrace{\begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}_{\text{K elements with } k\text{:th element = 1}}{}^T. \tag{6}$$

Let $\mathscr{C}$ be the set of prespecified classes used in the reference map, in our case $\mathscr{C} = \{\text{grass, asphalt, house}\}$. Obviously we need to be able to deal with the fact that the classifier can encounter objects unknown to it, e.g. due to occlusion or model imperfections. Let us define $S(i)$ to be the true "class" of superpixel i, in an abstract sense where we consider all thinkable classes. We can then only rely on our classifier in the case $S(i) \in \mathscr{C}$. The underlying class in the reference map, of the area captured by superpixel i, is modelled as another stochastic variable $\tilde{X}_{i,t}$ according to

$$\tilde{X}_{i,t} = \begin{cases} X_{i,t} & \text{if } S(i) \in \mathscr{C} \\ X_i^0 & \text{otherwise} \end{cases} \tag{7}$$

where X_i^0 is a default[1] value for $\tilde{X}_{i,t}$. Hence, if the image from the on-board camera for instance is occluded by some object unknown to the classifier, a default value is used instead of the value derived from the image. We will of course never know whether this is the case or not, but we can estimate the level of certainty in the classification

$$a_i \triangleq P(S(i) \in \mathscr{C}). \tag{8}$$

How this is done is described in Sec. II-C. By the law of total probability we can write

$$\begin{aligned} P(\tilde{X}_{i,t} = C^k) &= P(S(i) \in \mathscr{C}) P(X_{i,t} = C^k) \\ &\quad + P(S(i) \notin \mathscr{C}) P(X_i^0 = C^k) \\ \Rightarrow \tilde{X}_{i,t} &= a_i X_{i,t} + (1 - a_i) X_i^0. \end{aligned} \tag{9}$$

We can easily derive the expected value of $X_{i,t}$

$$\mathbb{E}[X_{i,t}] = \sum_{k=1}^{K} C^k p_i^k = p_i \tag{10}$$

and its covariance

$$\Sigma_i \triangleq \mathrm{Cov}(X_{i,t}) = \begin{pmatrix} \Sigma_i^{11} & \dots & \Sigma_i^{1K} \\ \vdots & \ddots & \vdots \\ \Sigma_i^{K1} & \dots & \Sigma_i^{KK} \end{pmatrix} \tag{11}$$

where

$$\begin{aligned} \Sigma_i^{kl} &= \mathbb{E}\left[(X_{i,t}^k - p_i^k)(X_{i,t}^l - p_i^l)\right] = \mathbb{E}[X_{i,t}^k X_{i,t}^l] - p_i^k p_i^l \\ &= \Big/ P(X_{i,t}^k = 1) = p_i^k, \quad P(X_{i,t}^k X_{i,t}^l = 1) = \begin{cases} 0 & \text{if } k \neq l \\ p_i^k & \text{if } k = l \end{cases} \Big/ \\ &= \begin{cases} -p_i^k p_i^l & \text{if } k \neq l \\ p_i^k - (p_i^k)^2 & \text{if } k = l. \end{cases} \end{aligned} \tag{12}$$

The default variable X_i^0 is assumed to be normally distributed with a mean (p^0) corresponding to the average class proportions in the reference map, and a large covariance (Σ_{X^0}) to reflect the fact that when X_i^0 is used it is nothing but a blind guess.

Once we have obtained the variables for each superpixel, we can calculate a probabilistic histogram for each circular region. To keep the notation simple, assume that we are only dealing with one circular region and remember that the following procedure applies to all of them. Let μ_i be the proportion of superpixel i in the circular region. The histogram will then become

$$Y = \sum_i \mu_i \tilde{X}_{i,t} = \sum_i (\mu_i a_i X_{i,t} + \mu_i (1 - a_i) X_i^0) \tag{13}$$

with expected value

$$\mathbb{E}[Y] = \sum_i (\mu_i a_i p_i + \mu_i (1 - a_i) p^0) \tag{14}$$

and covariance

$$\begin{aligned} \mathrm{Cov}(Y) &= \Big/ X_{i,t} \text{ independent of } X_j^0 \text{ and } X_i^0 \text{ i.i.d. } \forall i \Big/ \\ &= \mathrm{Cov}\left(\sum_i \mu_i a_i X_{i,t} \right) + \sum_i (\mu_i (1 - a_i))^2 \Sigma_{X^0} \\ &= \sum_i (\mu_i a_i)^2 \Sigma_i + 2 \sum_{i<j} \mu_i \mu_j a_i a_j \mathrm{Cov}(X_{i,t}, X_{j,t}) \\ &\quad + \sum_i (\mu_i (1 - a_i))^2 \Sigma_{X^0}. \end{aligned} \tag{15}$$

[1] X_i^0 is indexed with i to indicate that we have one instance of X^0 for each superpixel i, but all of them are independent and identically distributed (i.i.d.).

In (15), all terms are known except $\mathrm{Cov}(X_{i,t}, X_{j,t})$. All these cross covariances are in our current implementation assumed to be zero. In one sense this choice seems to be well motivated, due to the fact that all superpixels consist of uniform parts of the image with distinct borders between them. One could therefore assume that the classes of different superpixels should be uncorrelated. However, since the coarseness of the segmentation algorithm is controlled via user set parameters it is clear that this actually cannot be the case. One uniform region could very well be split into two superpixels if the algorithm was set to work with a finer segmentation, and these superpixels are then clearly not uncorrelated. In future work on this topic we intend to model the cross correlation using information about how strong the evidence is for a border between two superpixels. This information is already available from the segmentation algorithm [4].

It is intractable to derive the true distribution of Y, but since it is a sum of several stochastic variables we approximate it with a normal distribution for which we know the 1st and 2nd order statistics. To aquire the measurement equation from (2), Y is divided into a measurement which is the expected class histogram derived from I_t, $y(I_t, \hat{z}_t) = \mathbb{E}[Y]$, and a noise representing the uncertainty in the classification $e_t \sim \mathrm{N}(0, R)$, $R \triangleq \mathrm{Cov}(Y)$. This yields the sought measurement equation $y(I_t, \hat{z}_t) = h(x_t) + e_t$, where $h(x_t)$ is the precomputed class histogram from the look-up table. We can now, for instance, obtain the measurement likelihood as $p(y_t|x_t) = p_e(y_t - h(x_t))$. However, due to the special structure in the stochastic histogram ($||Y|| \equiv 1$), the probability density function (PDF) will be constrained to a $(K-1)$-dimensional hyperplane as shown in Fig. 4. This also means that R will be singular. To handle this we can make a coordinate change to coordinates that are local to the hyperplane, e.g. by utilizing a singular value decomposition of R. Given that $h(x_t)$ lies in the same hyperplane (which always will be the case since it is a histogram) we can compute the PDF as

$$p_e(y_t - h(x_t)) = \frac{(2\pi)^{-(K-1)/2}}{\lambda_1 \dots \lambda_{K-1}} e^{\left(-\frac{1}{2}(y_t - h(x_t))^T R^{\dagger} (y_t - h(x_t))\right)} \tag{16}$$

where, λ_k, $k = 1, 2, \dots, K-1$, are the $K-1$ non-zero eigenvalues of R and † is the Moore-Penrose pseudo-inverse. How the likelihood is computed from the singular normal distribution is illustrated in Fig. 4 for $K = 3$.

Once the individual likelihoods from all circular regions are obtained, the total likelihood for the measurement is given as their product. The algorithm is summarized in Alg. 1.

C. Classification Reliability

To obtain a robust geo-referencing system we need to deal with the fact that the classification can fail totally from time to time. This will for instance happen if the image becomes occluded or if the classifier encounters some unknown object. Recall from (8) that we have defined mixing coefficients a_i that in some sense can be interpreted as probabilities that the classification is reliable. To estimate these coefficients, we make use of the unnormalized class probabilities L_i^k from (4). A good classification is typically characterized by

$$L_i^k \approx \begin{cases} 1 & \text{if } k = \tilde{k} \\ 0 & \text{if } k \neq \tilde{k} \end{cases} \tag{17}$$

for some class $\tilde{k} \in \{1, \dots, K\}$. We therefore define a_i to be a linear interpolation of L_i in a K-dimensional hypercube, where the corners along the axes are assigned the value 1 and all other corners are assigned 0. For example, with $K = 2$ we define $a(0,0) = 0$, $a(0,1) = 1$, $a(1,0) = 1$, $a(1,1) = 0$ and interpolate to get $a_i = a(L_i^1, L_i^2)$. This method assigns a high value to a_i if one of the L_i^k:s is close to 1 and the remaining L_i^k:s are close to 0, in accordance with the objective (17).

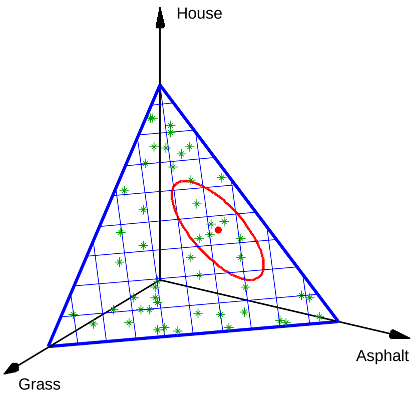

Fig. 4. A class histogram from three classes will be a point on the unit simplex in $\mathbb{R}^3$. Y is a stochastic variable in the same plane as the simplex, with mean and covariance illustrated by the dot and ellipse, respectively. Each point in the reference map has a histogram $h(x_t)$ on the simplex associated with it, illustrated by the asterisks (*). The likelihood is given by $p(y_t|x_t) = p_e(y_t - h(x_t))$.

Algorithm 1 Likelihood Computation

```
1) Segment the on-board image $I_t$ into superpixels.
2) For each superpixel $i = 1, \dots, M_t$:
    a) Extract a descriptor $d_i$ according to Sec. II-A.
    b) Feed $d_i$ to the classifier to obtain the unnormalized class probabilities, $L_i^k$.
    c) Compute $a_i$ from (8) according to Sec. II-C.
    d) Normalize $L_i$ to obtain $p_i$.
3) For each circular region in the image:
    a) Compute the 1st and 2nd order statistics of $Y$ from (14) and (15) using (12).
    b) Compute $p(y_t|x_t) = p_e(y_t - h(x_t))$ from (16) where $h(x_t)$ is taken from a look-up table.
4) Multiply the likelihoods from all circular regions to obtain the total likelihood.
```

D. Performance Analysis

Using the reference map shown in Fig. 1 and the classification result from Fig. 3, the resulting likelihood over the map according to Alg. 1 is illustrated in Fig. 5. We see that the likelihood is high in regions where we have both asphalt and houses in the reference map, since this is the case for the classified image. Along the roads, where the reference map consists of asphalt and grass but no houses, the likelihood is lower but still significantly more than zero. This is desired, since the houses in the classified image very well could be incorrectly classified. Finally, in regions where the reference map solely consists of grass, the matching is very poor and the likelihood is close to zero.

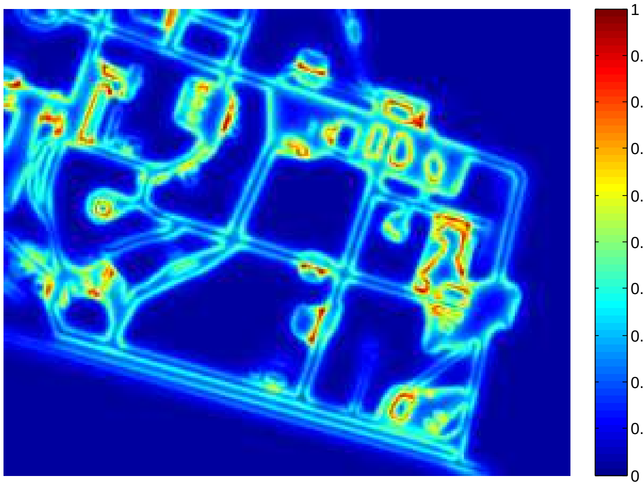

Fig. 5. Computed likelihood over the reference map.

III. Navigation framework

To test the geo-referencing it is implemented in a VO framework previously studied by Törnqvist et al. in [9]. The framework is briefly presented here, to show how it can be fused with the proposed geo-referencing, but for all details we refer to the original paper. In this application example, the vehicle state consists of position $x_{p,t}$, velocity $x_{v,t}$, acceleration $x_{a,t}$, a quaternion $x_{q,t}$ representing the orientation of the UAV and its angular velocity $x_{\omega,t}$. The state vector is also augmented with bias states for acceleration $b_{a,t}$ and angular velocity $b_{\omega,t}$ to account for sensor imperfections.

The navigation problem has a linear and Gaussian substructure which is exploited by using the marginalized particle filter (MPF) framework [7]. Hence, the state vector is divided into linear states x_t^l and nonlinear states x_t^n,

$$\begin{aligned} x_t^l &= \begin{pmatrix} x_{v,t}^T & x_{a,t}^T & b_{\omega,t}^T & b_{a,t}^T & x_{\omega,t}^T \end{pmatrix}^T, \\ x_t^n &= \begin{pmatrix} x_{p,t}^T & x_{q,t}^T \end{pmatrix}^T. \end{aligned} \tag{18}$$

VO is incorporated into the estimation problem by tracking a set of landmarks $m_t = \{m_{j,t}\}_{j=1}^{J_t}$ in consecutive frames. The landmark positions in an absolute coordinate system are also part of the linear state vector. The dynamic model of the system is

$$\begin{aligned} x_{t+1}^n &= f^n(x_t^n) + A^n(x_t^n)x_t^l + G^n(x_t^n)w_t^n \\ x_{t+1}^l &= \qquad\qquad\quad A^l(x_t^n)x_t^l + G^l(x_t^n)w_t^l \\ m_{j,t+1} &= m_{j,t}, \qquad j = 1, \ldots, J_t \end{aligned} \tag{19}$$

where w_t^n and w_t^l are assumed white and Gaussian with zero means.

Landmarks are initiated from distinct Harris corners in the on-board images and tracked between frames using NCC. This gives rise to a measurement, available at 4 Hz (the image frequency)

$$y_{\text{VO},t} = h_{\text{VO}}(x_t^n) + H_{\text{VO}}(x_t^n)m_t + e_{\text{VO},t}. \tag{20}$$

The vehicle is also equipped with an IMU and a barometric sensor, working at 20 Hz, yielding a second measurement

$$y_{\text{IMU},t} = h_{\text{IMU}}(x_t^n) + H_{\text{IMU}}(x_t^n)x_t^l + e_{\text{IMU},t}. \tag{21}$$

The measurement noises $e_{\text{VO},t}$ and $e_{\text{IMU},t}$ are assumed white and Gaussian with zero means.

Finally, we have a third measurement from the geo-referencing according to (2). This measurement enters the filtering scheme in the computation of importance weights used for particle resampling. For N_p particles, the importance weights $\gamma_t^{(i)}$, $i = 1, \ldots, N_p$, are proportional to the measurement likelihood

$$\begin{aligned} \gamma_t^{(i)} &\propto p(y_{\text{VO},t}, y_{\text{IMU},t}, y_t | x_{1:t}^{n,(i)}) \\ &= p(y_t|x_t^{n,(i)})p(y_{\text{VO},t}, y_{\text{IMU},t}|x_{1:t}^{n,(i)}) \end{aligned} \tag{22}$$

where the second factor is derived in [9] and the first factor is given by Alg. 1.

IV. Experimental results

This section presents experimental results for UAV navigation using the MPF approach presented in Sec. III. Data used in the experiments was collected during a 400 m test flight in southern Sweden, using an unmanned Yamaha RMAX helicopter. Fig. 6 shows a map over the area with the UAVs true flight trajectory (a Kalman filtered GPS signal) illustrated with circles.

The horizontal position from the VO solution from [9] is plotted as a dashed line. We can see that the estimate is fairly accurate, but as expected it suffers from a drift. In the same plot, also the solution using both VO and geo-referencing is shown as a solid line. The estimated trajectory in this case is very close to the ground truth, and it seems as if the drift has been removed.

In Fig. 7 the error in horizontal position is shown. The error has been divided into two components. The first is the error orthogonal to the road over which the UAV is flying and the second is the error along the direction of the road. The orthogonal error is much smaller when the geo-referencing is used and the drift is completely removed. The error parallel to the road on the other hand has not been reduced significantly by the geo-referencing, and the drift is still present.

The reason for this is that the geo-referencing is much less informative in the direction parallel to the road. If the UAV is flying along a road with grass on both sides it is obvious that it will not be able to know exactly how far it has flown. Compare with a human driving along a road. We usually have a very accurate estimate of our orthogonal position, namely that we are on the road, but we do not know exactly how far we have driven. However, as soon as we encounter a distinct landmark, such as a crossing or a house, this information is inferred to allow for an accurate estimate of our position along the road as well. It is desired that this should be the case also for the geo-referencing system, but from Fig. 7 we

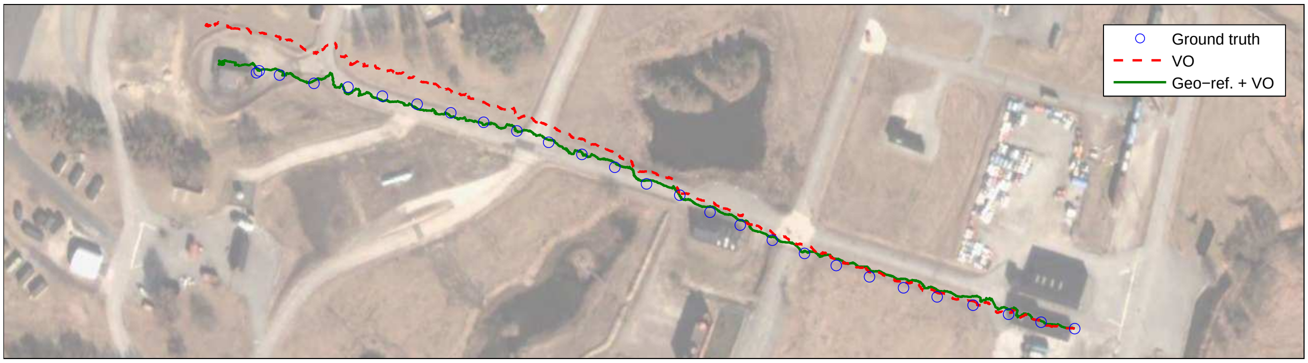

Fig. 6. True trajectory illustrated with circles and the estimated trajectories with (solid line) and without (dashed line) geo-referencing.

see that it is not. Our experiments indicate that there are two major reasons for why the geo-referencing fails in this sense.

The first is that the matching procedure has been made rotation invariant to cope with instability issues. As pointed out i Sec. I, this means that some information is discarded. Take for instance the case when the UAV encounters a crossing. Since the geo-referencing uses class histograms from circular regions this will not be seen as a distinct crossing by the system, but merely as if the proportion of road increases. This is a drawback with the proposed system, and further investigation of the tradeoff between stability and estimation accuracy is required.

The second reason for the lost accuracy is the treatment of classification uncertainty, as described in Sec. II-C. Due to the possibility of misclassification the system will never fully rely on what is seen in the image, which also is a tradeoff between accuracy and robustness. Here the performance could be increased by improving the classification and/or the outlier rejection process. This is something that we intend to do in future work, and we will then hopefully be able to deduce which one of these two reasons that is most responsible for the lost estimation accuracy.

V. Conclusions

A geo-referencing system for absolute UAV positioning has been developed. The position reference is expressed as a standard measurement equation, making it easy to incorporate into any sensor fusion framework. The system makes use of environmental classification and rotation invariant template matching, making it robust to variations in the operational environment as well as errors in the estimated orientation of the vehicle. Any probabilistic classifier can be used together with the proposed geo-referencing system. The measurement model is available as a 2D look-up table with additive noise. The noise distribution is derived from the classification result, reflecting the uncertainty in the classification.

The system is shown to significantly improve the estimation accuracy in directions where the measurement model is rich on information. However, in directions where the measurement model is low on information the system fails to remove the drift in the position estimate. Further research is needed to be able to improve the performance in this sense, while still maintaining a high level of robustness.

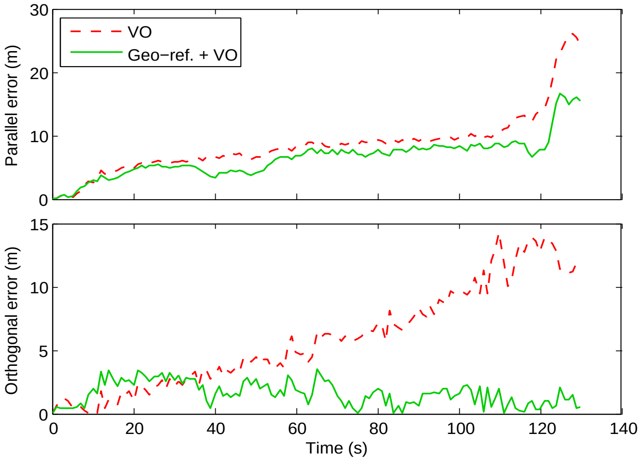

Fig. 7. Error in horizontal position estimate in the direction parallel to the road (top) and orthogonal to the road (bottom).

Acknowledgement

This work was supported by the Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research, SSF and CADICS, a Linnaeus center funded by the Swedish Research Council.

References

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *Robotics & Automation Magazine, IEEE*, 13(3):108–117, Sept. 2006.

[2] G. Conte and P. Doherty. Vision-based unmanned aerial vehicle navigation using geo-referenced information. *Accepted for publication in the EURASIP Journal of Advances in Signal Processing*, 2009.

[3] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): part I. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, June 2006.

[4] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *Intl. Journal of Computer Vision*, 59(2):167–181, 2004.

[5] J. Kaufhold, R. Collins, A. Hoogs, and P. Rondot. Recognition and segmentation of scene content using region-based classification. In *Proc. of the IEEE Intl. Conf. on Pattern Recognition*, 2006.

[6] I. Posner, M. Cummins, and P. Newman. A generative framework for fast urban labeling using spatial and temporal context. *Autonomous Robots*, 2009.

[7] T.B. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Trans. Signal Process*, 52(7):2279–2289, Jul. 2005.

[8] D. G. Sim, R. H. Park, R. C. Kim, S. U. Lee, and I. C. Kim. Integrated position estimation using aerial image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):1–18, 2002.

[9] D. Törnqvist, T.B. Schön, R. Karlsson, and F. Gustafsson. Particle filter SLAM with high dimensional vehicle model. *Journal of Intelligent and Robotic Systems*, 55(4):249–266, 2009.