

A GOOGLE-Earth Based Test Bed for Structural Image-based UAV Navigation

Eckart Michaelsen

Target Recognition Department
FGAN-FOM
Ettlingen, Germany.
mich@fom.fgan.de

Klaus Jaeger

Target Recognition Department
FGAN-FOM
Ettlingen, Germany.
jae@fom.fgan.de

Abstract - In this contribution a test bed is presented for the assessment of structural recognition methods for the localization of landmark objects in aerial images. Instead of really flying a system which might jeopardize the utilized platform – or using only a limited test image data set which is of poor relevance – the recognition system is included into a control loop with the camera being simulated by a publicly available geo system such as GOOGLE earth. The structural knowledge is represented as declarative production system. An approximately correct any-time parser is used for controlling the search. A typical example run is discussed. Such test bed can help evaluating and improving the performance of such systems for the task of autonomous visual UAV navigation.

Keywords: UAV navigation, evaluation, structural recognition, production systems, public geo systems.

1 Introduction

Automatic fusion of declarative knowledge about e.g. infra structure construction with measured pictorial data e.g. aerial images is a challenging endeavor. If successful it may well be of great advantage for tasks such as autonomous air navigation. In a way the ultimate goal should be to give the UAV a kind of understanding of the scene below it. For the time being we reckon it sufficient if we can give an automatic method for recognizing specific salient landmarks. Here we distinguish *declarative knowledge* such as handbooks or thesauri on the construction of salient infra structure such as bridges, major traffic ways, power plants etc. from *pictorial knowledge*. The latter consists of large representative bodies of aerial pictures of such objects. Given such data the landmark recognition could be achieved using standard appearance based object recognition methods. Whatever those are – the chink of them rests in the word “representative”. In outdoor scenery lighting and the brightness of background objects such as plants varies dramatically with daytime and season. The reflectivity of the visible surfaces of the landmark objects – such as concrete or asphalt – varies also considerable with age humidity etc. The geometry of shadows cast by 3D structure depends heavily on the time of day and cloud

cover. The camera will usually be equipped with automatic exposure control – so that a particular gray-value does not correspond to any calibrated brightness. And last but not least – unpredictable arbitrary clutter objects may be present on the landmark and on the background.

On the other hand, while a structural recognition method based on declarative knowledge can be *implemented* without any image material at all it cannot be *evaluated* so. And using the same image material that has been used for development and debugging is also questionable. The evaluation should be done using images that did not even exist when the method was fixed.

1.1 Related work on UAV navigation

Precision navigation of unmanned aerial vehicles (UAVs) like missiles and drones is still one of the most important subjects in defense research. Current system design uses inertial measurement units (IMU) and/or GPS for UAV guidance. However, due to the fact that inertial data are affected by noise or drift and GPS-data may be jammed by countermeasures, accurate measurement of missile pose with respect to the ground scenario may be difficult. To overcome these problems, cutting-edge system designs are specified by concepts of hybrid navigation, fusing in real time all available navigation data such as IMU/GPS, radar altimeters, star tracker, passive imaging sensor and digital elevation database [1]. The integration of passive imaging sensors has some important advantages: they can operate in non-GPS environment and in scenarios with poor variations of surface elevation. Hence image based navigation of UAVs is common topic for decades now and there are many approaches for computing ego-motion from optical flow measurements (e.g. in [2]) and Structure from Motion (SFM) algorithms – such as projective geometry and bundle adjustment (e.g. in [3]). In this paper, we propose a method for navigation update based on the automatic recognition of suitable landmarks by analyzing image data of high resolution (IR, Visible, SAR).

1.2 Related work on object recognition

For many years now the majority of the object recognition literature evaluates the approaches on turntable benchmark sets such as COIL [4]. Turntable data are particularly useful in the evaluation of 3D-object recognition. Often, statistical modeling (i.e. optimization of MAP) turns out to perform best. For the task of autonomous UAV navigation 3D recognition capabilities may be only of interest if it is intended to fly at very low altitudes. For medium and high altitude navigation the errors caused by 3D-structure – such as occlusion of lower parts by higher ones – or the respective displacements – e.g. caused by moderate tilting of surfaces or different scaling due to different height – should be no problem for a robust and error tolerant recognition system as the one used here.

However, turntable data usually lack background and variation of lighting. There are attempts to alleviate the resulting lack of robustness of the statistical methods – e.g. [5]. Still, for aerial images taken in a different season under completely different lighting this will not suffice.

1.3 Related work on structural aerial image understanding

The archetype for structural aerial image understanding remains the SIGMA system [6]. Another example for such systems which were quite popular two decades ago was the SCHEMA system [7]. One reason why these kind of systems are not so popular anymore may be the lack of possibility for objective evaluation – such as is given by tests with benchmark data as outlined in Section 1.2. Still, our own work stands in continuation of these classics. In Section 2 we give a production system and a corresponding interpreter and control structure. We have used such systems e.g. for fusion of evidence from different sensors in a structural and knowledge-based manner [8]. And we hope that the work published here may help mitigating the evaluation problem.

The issue of incorporating knowledge about the standard infrastructure in the region where the task is to be performed into the recognition system is treated e.g. in [9]. There the AI-system ALFIE was proposed in which the feature extraction methods are chosen and parameterized according to the geographical context emphasizing that this is non-trivial and important. Road extraction was the major application - in contrast to our landmark recognition, and satellite images where the main data source in contrast to aerial images in our work. Still, our argument is along the same lines.

2 Structural landmark recognition

The idea here is utilization of known constructive features and patterns of salient man-made objects. Instead of relying on learning data the system uses knowledge sources such as handbooks on infrastructure construction, thesauri, or even

Wikipedia. The hope is of course that the performance of such systems will not depend on whether the learning data are representative but on the quality of the utilized knowledge and its correct and suitable representation – which can be improved almost arbitrarily by diligent labor.

2.1 An example object and its representation

Let us take a look at the example of a simple bridge crossing a German standard Autobahn. Every few kilometers there is a road or railway crossing along such an autobahn. And - in Germany - the autobahn-net is dense enough that such a bridge is not far away from any point in country. It is known that autobahns have a standardized lane width of 3.3 meters. Most common design is triple lane or dual lane with and additional margin called “Standspur”. For the upper part of the bridge we have larger variation. But all in all we can summarize the knowledge of such an object in the following statements:

A *bridge* is an object consisting of two parts: A *road_stripe* and a *highway* where the upper part is occluding the lower part when viewed from above. The *road_stripe* is made of different material than its surroundings such that very often it causes a straight long contour on both margins in aerial images. Simplified, we may state that a *road_stripe* causes two parallel *long_line* structures in the image. We do not know whether the road will be brighter or darker than the background – in fact the direction of contrast may flip anywhere, e.g. due to surface renewal. But we may state that the contours of a road are in good continuation: A *long_line* is made of a considerable number of contour primitive objects *line* which are aligned and overlapping or with minor gaps in continuation.

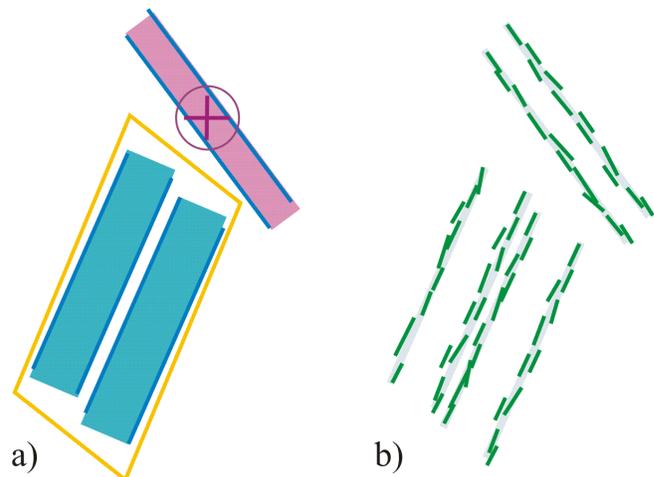


Figure 1. Knowledge about the anticipated appearance of a bridge in an aerial image

A *highway* is composed of two parallel *highway_stripe* objects which lie a certain distance apart. These are similar to *road_stripe* objects – only a little wider and with less

tolerance in width. In Fig. 1 some of this knowledge is presented in a pictorial manner using the same color which is applied in the text to the symbols representing the entities of this discourse. Fig. 1.a) depicts a possible mutual geometric positioning of the non-primitive objects. In Fig. 1.b) a suitable set of primitives is given supporting this layout. Arbitrary other primitive and non-primitive objects in the picture would be regarded as clutter.

The presented declarative knowledge about such a landmark connects the concept “bridge over an Autobahn” with measurements that can be easily obtained from an image by using a gradient filter and a threshold (as they are provided by most image processing tool boxes). Here we sketched this knowledge informally. Errors and misunderstanding can of course be avoided when AI-formalisms or other well defined structures are utilized – such as UML. But in principle no example images are needed. While advancing to a more precise formulation of the knowledge the parameters inherent in the geometric constraints have to be fixed. Most of these can be set from available sources – such as the set value and tolerance for widths of the objects and angle tolerances. An overall scaling can be estimated from the flight elevation and the focal length of the camera. Other tolerances – such as the deviation in position and orientation of the primitive *line* objects from the corresponding non-primitive *long_line* object can be obtained from the properties of the image filter used for the object orientation. For the setting of these parameters some representative image data are actually helpful, in particular to estimate the noise in homogenous regions and distribution of gradient magnitudes at road margins. With no image material at all one may still take the default values recommended in the image processing software manual.

2.2 The production system formalism

In order to utilize a piece of declarative knowledge – such as developed in Section 2.1 – in a machine it has to be coded accordingly. A convenient way is given by production systems. The example gains the following form:

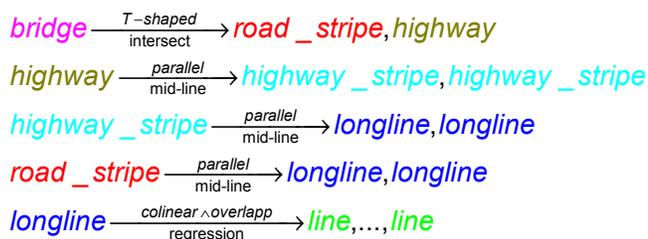


Figure 2. Example production system, the colours of the symbols are also used in other figures and in the text

As usual for context free systems there is always only one non-terminal symbol on the left-hand side of the productions. According to the right-hand side we distinguish two normal forms: 1) An ordered pair of the

same or different symbols (such as numbers 1 to 4); 2) a set from instances of the same symbols. These two forms are treated differently when parsing an image as indicated below in Section 2.3. In contrast to string rewriting grammars this formalism works on sets of objects that have geometric attributes (such as locations and orientations in the image). An elaborated theory for such structures can be found in [10]. In our notation the constraints (that replace the concatenation used in string grammars) are written above the production arrow. Such constraint-set-grammars also need the definition of a function constructing the attributes of the left-hand side for each production (here written below the arrow). Syntactically, the system can be used left to right. Then it defines a language – the set of all images that contain such a bridge. Using a random generator it can be utilized as computer graphics tool to construct example images. This direction is called the *generative* direction. The other direction – right to left – is called the *reducing* direction. This direction is parsing given input images for the presence of such objects. Due to the combinatorial nature of the definitions sound and complete parsing may be infeasible. We will give approximate solutions in the next section.

2.3 Approximate any-time parsing

In order to utilize a piece of declarative knowledge – such as developed above instances of the most primitive object type have to be extracted (i.e. segmented) from the image. Figure 3 shows the pseudo-code of the hypotheses driven parser that is used. Basically, every object instance *new_elem* causes the construction of a hypothesis – first it is a *nil* hypothesis. The administration of the hypotheses in the queue is the main process. Several of these may be processed in parallel threads (in the example in Section 2.4 64 parallel hypotheses were processed). If they are *nil*-hypotheses they will be cloned according to the right-hand side of the productions. For instance in the system given in Section 2.2 a *nil*-hypothesis corresponding to a *long_line* object will be cloned twice once as hypothesis for a *road_stripe* object and once as hypothesis for a *highway_stripe* object. Else, if they have a production associated they will trigger a query for corresponding partners that fulfill the constraint. It is evident from the pseudo-code that for productions of the normal form 1) a combinatorial search is performed. A nested for loop will often cause the construction of multiple new object instances. For this particular system a polynomial complexity of sixth order can be predicted in the number of *long_line* objects. Other such systems that contain recursive loops (i.e. real syntactic structure) in their normal form 1) productions are of exponential complexity. Thus a sound parser is not robust and feasible for such systems. It is hard to predict the computational effort for a complete search – it may be critically dependent on the input data and the thresholds. There are two ways to mitigate this disadvantage: First one may abstain from complete search.

Instead, the hypotheses are occasionally sorted according to priorities. These may result from the quality of the corresponding objects (bottom up); or they may result from the importance of the production and from a focus of interest set by the current search situation (top down). Different top-down strategies for systems of this sort have been compared e.g. in [11]. In the example run documented in Section 2.4 such control strategies are used. Of course an incomplete search will only give an approximate parse. But the search can thus gain any-time capability.

```

while queue not empty OR other break do
  sort(queue);
  set_of_hypothesis = choose_best_n(queue);
  foreach trigger_hypo ∈ set_of_hypothesis do
    if p=nil then
      remove_queue(trigger_hypo);
      foreach q where trigger_obj ∈
        right-hand side do
        new_priority = prio(q) *
          priority(trigger_hypo);
        append_queue(trigger_elem, q,
          new_priority);
        end
      else
        actual_query =
          construct_query(trigger_hypo);
        candidate_set =
          pose_query(actual_query);
        if p of Normal_form_1 then
          foreach partner ∈ candidate_set do
            p:new_elem ← (trigger_elem,
              partner);
            add_database(new_elem);
            construct_null_hypo(new_elem);
          end
        else
          p:new_elem ← candidate_set;
          add_database(new_elem);
          construct_null_hypo(new_elem);
        end
      end
    end
  end
end
end

```

Figure 3. hypotheses driven control loop

Second one the normal form 2) for productions is introduced as a shortcut for clustering sub-systems [12]. In the example system the fifth production replaces a subsystem containing $long_line \rightarrow line, line$ and $long_line \rightarrow long_line, line$. As can be seen in the pseudo-code in the lower internal *else* branch, such shortcut productions of normal 2) are not treated in the sound combinatorial way. Not all possible sub-sets fulfilling the constraint are enumerated. Only the maximal set fulfilling the constraint is chosen. Of course this is also an approximate solution.

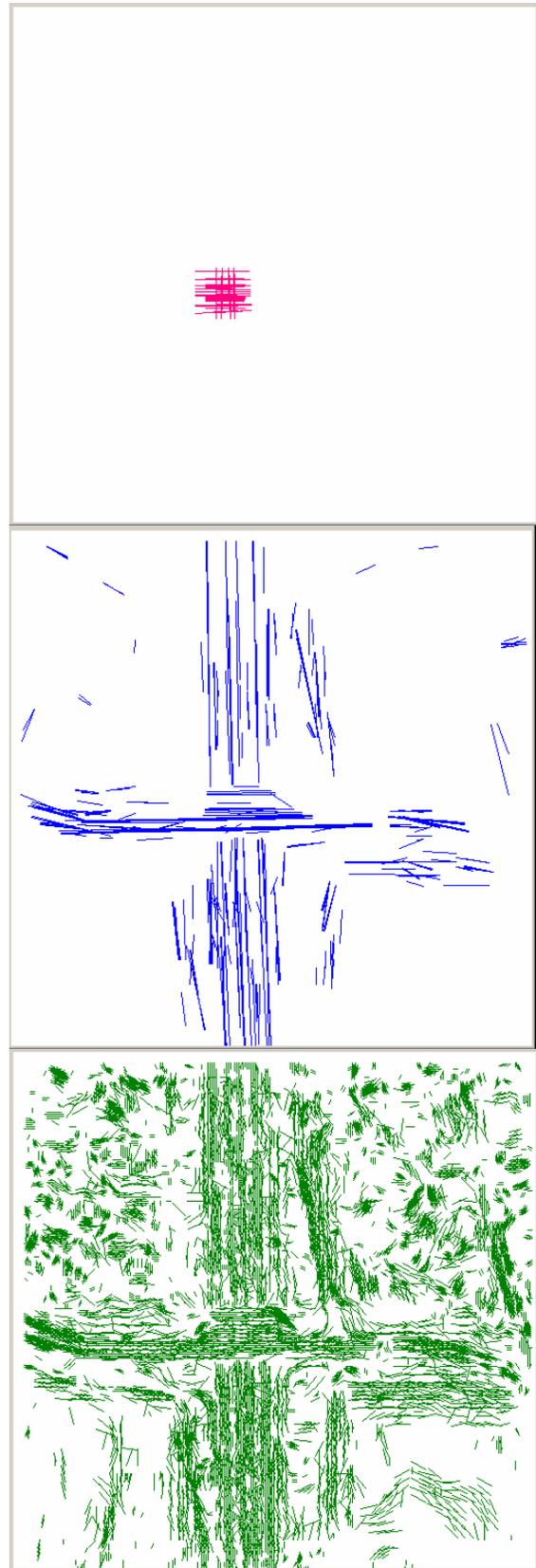


Figure 4. Results on one example image

2.4 One example run

In Figure 4 a result is displayed that has been achieved using the production system given in Section 2.2 with the control indicated in Section 2.3. Break criterion was here at least ten instances of the object *bridge* found or the exceedance 60 seconds of overall search time. A rather typical image of a German autobahn with a bridge and a village nearby was used. The feature extraction – a squared averaged gradient filter with threshold on the ratio of the two eigenvalues – is run on scaled versions of 256^2 , 128^2 , and 64^2 pixels respectively, resulting in 14837 *line* object instances which are displayed in the lower part of Figure 4. Actually in this run 52 *bridge* instances were reduced in approximately 4 seconds calculation time on our server (which is a machine operating 8 CPUs at 3GHz, and we are picking always the 64 best hypotheses from the queue running the system also in 64 threads). Between these root objects and the primitives we have 66 *road_stripe*, 64 *highway_stripe*, 20 *highway*, and 292 *long_line* object instances. The latter are displayed in the center of Figure 4.

The resulting landmark locations displayed in the upper part of Figure 4 cluster roughly around the correct position. However, there is a bias to the North. This is a frequent – and foreseeable - displacement when operating the system in bright sunlight at noon on the northern hemisphere: The shadow of the bridge will often be mistaken for the object itself. Still the center of gravity of the 52 *bridge* instances is located 38m West and 16m South of the image center which is a good correction – such that the next image at the next landmark will also actually contain this landmark object and the flight simulation will not go astray.

3 The test bed

A structural recognition system – such as the one presented in Section 2 leaves a lot of room for improvement and tuning. Also, though being constructible without example data, it can only be evaluated using representative image data. And moreover, input data are needed to debug the system. Thus it is a non-trivial task to assess the value of such systems for automatic landmark-based aerial navigation. We see three possible approaches for this:

1) One way to provide suitable data would be to equip an unmanned aerial vehicle with a camera and feed the incoming images to the system. The decisions and measurements of the system would then in turn be used as input to the flight control system of the platform. This would be the test closest to the application meant – the gold standard. However, next to many technical problems and high costs – in particular in the case of recognition failures which may of course lead to the loss of the platform – it also poses severe judicial problems: The use of autonomous aerial platforms is very restricted in civilized areas and who should be liable in case major damage is caused?

2) Therefore most often the data were acquired in a different way: Aerial images of interesting landmarks where taken in measurement campaigns or bought from commercial providers. With a sufficiently large set of such images the system could be debugged, tuned, and assessed. It was possible to compile statistics from which a success rate and a mean square error could be estimated. However, this requires discipline from the developers: In order to be significant, the data should be separated into a learning set used for debugging and tuning and a test set used for assessment. The developers should not consult the test set during the learning phase, and nothing should be changed anymore in the assessment phase. Best way to guarantee this is to take the test pictures after the system has been fixed and let the assessment be done by another working group. Also the test pictures should not be taken from the same region, or in the same season and daytime etc. Moreover, this assessment evaluates only an isolated component of the landmark navigation control cycle. In the control loop the position from which the $n+1$ th picture is taken is dependent on the result of the recognition and measurement in the n th picture. After all recognition rate and precision of the recognition system are not the desired estimation parameters. Instead, the probability of going astray has to be estimated – in particular in a fusion setting, where the navigation is not only landmark based but also includes an inertial system and maybe other means. Inferring such probability from the recognition rate and precision may not be easy and error prone.

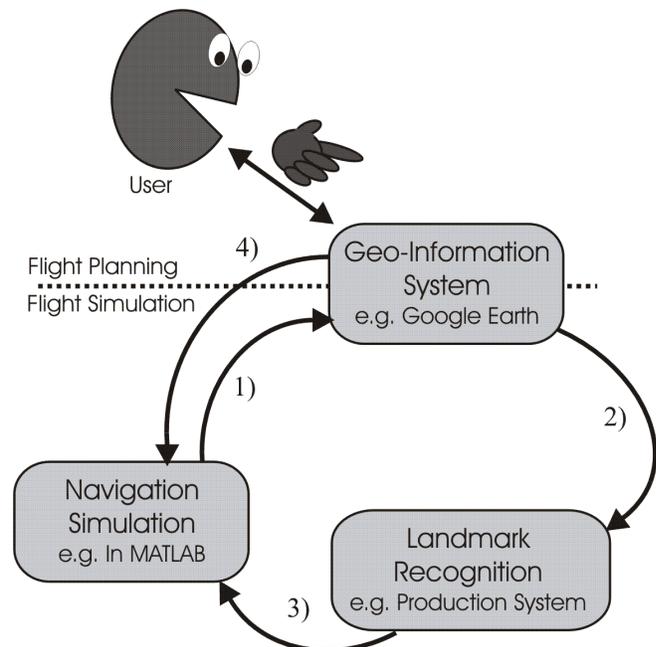


Figure 5. Schema of the test bed

3) A compromise today is to use one of the publicly available geo systems such as *Google Earth* or *MS Live Search* as camera simulator. This approach is sketched in Figure 5. Next to the geo system and the landmark

recognizer a third system is required. We call it the navigation simulator. It simulates the flight and the other navigation systems that are fused with the landmark recognition. It uses a random generator. The main advantage of this approach is – it closes the control loop while avoiding the costs and risks of a real flight.

In Figure 5 four interfaces are indicated by numbers: 1) the navigation simulator gives a specification to the geo system that is sufficient to acquire an image. This is essentially a geo location (in North and East as angles in degree Minutes and seconds). These data may be augmented by an altitude or orientation angles in yaw, pitch, and roll respectively. 2) The geo system gives an image to the landmark recognizer. This may be augmented by auxiliary annotations, e.g. the focal length or height above ground or scale of pixel to meter, the principle point of the camera etc. 3) The landmark recognizer gives a position measurement to the navigation simulator. There may be an additional assessment about the reliability, certainty, or probability of this measurement and also an estimation of its error (a variance). This interface closes the loop such that the system can run automatically without interference of a user or developer. The evaluation measure then is simply whether and how often the system has lost its path. 4) A user has to specify the landmark path in a format accessible by the navigation simulator (say e.g. XML based KML files). If the user is identical with the developer he should only see the map layer of the geo system, but not the images – otherwise he/she would be tempted to pick landmarks that suit the system particularly well seeing images very similar to the ones that will be used in the loop. This would bias the performance assessment.

In our current implementation the navigation simulator sets the new actual position x_n from the old x_{n-1} basically by adding the difference of the setpoints d_n .

$$x_n = x_{n-1} + d_n + |d_n|(\beta + \varepsilon) - c_n$$

The correction c_n comes through interface 3). For simplicity these are currently all just planar vectors in meters. This formular models a Gaussian drift in two quantities: A standard deviation ε (currently set by default to 0.3%) and a bias β (currently set by default to 0.1% to the East). Thus after a flight distance of 10km the expected position of the landmark is 10m west and it the expected deviation from that is 30m. Thus, without correction, a flight will get lost after about a hundred km. We have made several test runs over German autobahns each using a dozen landmarks or so. For these experiments the recognition outcome is mostly like that indicated in Section 2.4: The landmarks found cluster densely round the correct position with a slight bias to the North. Each such outcome gives a proper correction in particular regulating against the bias. Occasionally, two or more clusters of landmarks are found. Most often the correct one is the one with more and

better objects – giving the same correcting result. Two kinds of errors also occur now and then (each with roughly 5% frequency or so):

1) No landmark is found within the time bound of 60 seconds. This happens often due to missing contrast. The modeled contours do not appear. This is a minor problem because in these cases the navigation simulator just continues with the prior estimation (landmark precisely in the middle of the image). Even if this would happen successively three times (which never occurred) the next landmark would probably still be in the image.

2) The best (or only) landmark cluster is wrong. Such error happens when adjacent structure – such as factory buildings or other traffic infrastructure – resembles the modeled structure. This causes a radical correction. The next landmark will be considerable off center. In all these cases it was found still – so that the flight can recover from this error. Two such errors in a row (and in the same direction) would be enough to lead the flight astray. But we never observed such behavior up to now.

4 Discussion

The presented work has somewhat preliminary character. There are many more questions posed than answered. Therefore, before concluding this contribution in Section 4.2 we will outline those directions of future corresponding research in this field which promise most benefit for the application.

4.1 Future work

The theory of productions systems as presented in Section 2 is not at all complete. E.g. Section 2.3 treats approximate correct parsing. Classically, a parse is either correct or not. “Approximation” here, first of all, requires a metric on partial parses, and then mathematical results are needed on the convergence of the algorithm to the correct complete parse according to this metric – and bounds on the residual incorrectness or risk of incorrectness respectively. The setting of the thresholds in the constraint part of the productions should be based on a normal distributed error model and a uniformly distributed clutter model – similar to the model introduced in [13]. At least the background clutter model should be learned and updated from intermediate images taken between the landmarks. Also, of course, many other salient landmark objects should be expressed as production systems – compiling a considerable catalog of objects.

We are aware that the flight simulator is also fairly crude yet. The error model should be replaced by one realistically modeling an inertial system. And Bayesian inference should replace the current simple acceptance of the best landmark position cluster. A Kalman filter should be included for the

simulation of the fusion of data from different simulated sources. The system has the setpoints, corrections, and actual points for each flight experiment and all example pictures. Not only counting the recognition rates is permitted but also estimating the quantitative deviation error for correct recognition cases. And this is possible for a real test set previously unknown to the developer – a rare and most favorable situation in pattern recognition! First results on the recognition rates and calculation times dependent on different control rationales will soon be published [14].

4.2 Conclusion

Previous application of structural landmark recognition was part of an automatic target recognition system for missiles and UAVs (unmanned combat air vehicles) [15]. Fusing data from a FLIR sensor and a real aperture radar (RAR) man-made structures like bridges were localized. Image sequences were taken by time-consuming and expensive measurement campaigns. Hence evaluation results were based on only a small set of different scenarios. On the other hand, the proposed GOOGLE-Earth based test bed offers the possibility of systematic evaluation of landmark recognition for UAV navigation independent from complex sensor data gathering. First experiments with the geo system in the simulated control loop indicate promising perspectives for the development and assessment of such knowledge based recognition systems. We conclude that the development of visual landmark-based navigation systems for UAVs – and in particular knowledge based such approaches – can benefit a lot from the utilization of such popular current Geo-systems. This certainly deserves further attention.

References

- [1] O. Aboutalib, B. Awalt, A. Fung, B. Thai, J. Leibs, T. J. Klausutis, R. Wehling, M. Iames, “All source adaptive fusion for aided navigation in non-GPS environment”, *2007 SPIE Defence & Security Symposium, Visual Information Processing XVI*, Z. Rahman; S. E. Reichenbach; M. A. Neifeld, (Eds.), Orlando FL, 8-13 April 2007, Proceedings Vol. 6575.
- [2] Z. He, R. V. Iyer, P. Chandler, “Vision-based UAV Flight Control and Obstacle Avoidance”, *Proc. IEEE American Control Conference*, Minneapolis, MN, pp. 2166–2170, June 2006.
- [3] D. Bulatov, “Towards Euclidean Reconstruction from Video Sequences”, *VISAPP 2008*, Proc. on CD, Vol. 2. pp 476-483, 2008.
- [4] <http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100>, last accessed March 2, 2009.
- [5] M. Grzegorzec, *Appearance-Based Statistical Object Recognition Including Color and Context Modeling*, Diss., Logos Verlag, Berlin, 2007.
- [6] T. Matsuyama, V. S.-S. Hwang. *SIGMA a Knowledge-Based Aerial Image Understanding System*, Plenum Press, New York, 1990.
- [7] B. Draper, R. Collins, J. Brolio, A. Hanson, E. Riseman, “The Schema System,” *IJCV*. Vol. 2. pp. 209-250, 1989.
- [8] E. Michaelsen, K. Jaeger, “Evidence Fusion Using the GESTALT-System,” *FUSION 2008*, Proc. on CD, IEEE Catalog Number CFP08FUS-CDR, pp 1147-1153, 2008.
- [9] S. J. Wallace, M. J. Hatcher, G. Priestnall, and R. D. Morton, “Research into Framework for Automatic Linear Feature Identification and Extraction,” in: E. P. Baltasvias, A. Gruen, L. Van Gool (eds.), *Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*, Balkema Publishers, Lisse, pp. 381-390, 2001.
- [10] K. Marroitt, B. Meyer (eds.), *Visual Language Theory*, Springer, Berlin, Heidelberg, New York, 1998.
- [11] E. Michaelsen, L. Doktorski, M. Arens “Making Structural Pattern Recognition Tractable by Local Inhibition”, *VISAPP 2009*, Proc. on CD, Vol. ... pp ..., 2009.
- [12] E. Michaelsen, L. Doktorski, M. Arens, “Shortcuts in Production Systems - A way to include clustering in structural Pattern Recognition”, *PRIA-9-2008 in Nischnij Nowgorod*, Lobashevski State Univ., ISBN 978-5-902390-14-5, Vol. 2, pp 30-38, 2008.
- [13] E. Michaelsen, U. Stilla, “Probabilistic Decisions in Production Nets: An Example from Vehicle Recognition”, in: T. Caeli et al. (eds.) *SSPR&SPR 2002*, LNCS 2396, Springer, Berlin, pp 225-233, 2002.
- [14] E. Michaelsen, M. Arens, L. Doktorski, “Interaction of Control and Knowledge in a Structural Recognition System”, submitted to KI-2009 in Paderborn, Germany, due September 2009.
- [15] K. Jaeger, K. Bers, K. Jurkiewicz, Three-Dimensional Intra- and Inter-Frame Target Detection in IR/mmW Image Sequences. In: Andresen B, Fulop G, Strojnik M (eds) *Infrared Technology and Applications XXVI, Proc. SPIE*, Vol. 4130, pp. 679-687, 2000.