Knowledge-based Recognition of Man-made Landmarks in a Simulated Control Cycle Using a Virtual-globe System

Eckart Michaelsen

Fraunhofer IOSB, Gutleuthausstrasse 1, 76275 Ettlingen, Germany

eckart.michaelsen@iosb.fraunhofer.de

ABSTRACT

Automatic knowledge-based recognition of landmarks in aerial images for UAV navigation is an alternative to GNSS navigation. It provides absolute position estimates thus complementing INS navigation. Relying on knowledge instead of template images or training samples is advantageous because the first may be out-of-date and the latter not representative. The robustness and precision of the method can be assessed using internet-based virtual globe systems such as Google Earth as camera simulator. This provides an almost open-world test-bed, in which the recognition system can be operated in a perception/action loop. The system proposed here has been elaborated for simulated flights over Germany using mainly highway bridges as landmarks. But its behaviour is also investigated over quite different regions. Of course also other landmarks such as churches or mosques can be used as landmarks.

1.0 INTRODUCTION

In the early days of aviation navigation was done exclusively using visual perception. With growing range of aircrafts this visual navigation was regarded as a challenge demanding high cognitive capabilities. Often a specialist officer was placed in a transparent nose cupola with good nadir and forward view. His main task was to understand where the aircraft was, by identifying landmarks on the ground. Since the advent of electronic navigation devices this specialist has lost his importance and he and his cupola vanished in contemporary designs. Also modern unmanned aerial vehicles (UAVs) – though they often have nadir looking cameras mounted – usually lack automatic understanding of landmarks. Their main information sources for navigation are dead reckoning on gyroscope evidence (INS) and global navigation satellite systems (GNSS) for absolute positions. But GNSS can easily be jammed today using cheap devices anybody may get hand on.

1.1 Automatic Visual Recognition for Aerial Navigation

The role of machine vision in the state-of-the-art aerial navigation is rather marginal. It is obvious that vision cannot be an option over the open sea or unstructured desert. This contribution focusses on populated areas with salient, large, and man-made structures. There, quite high precision can be expected already from rather primitive template matching. But this is only applicable when up-to-date templates are at hand precisely predicting the actual appearance of the objects in the moment they are used. In an uncontrolled outdoor environment under changing lighting, with changing seasons etc. there is doubt whether the templates resemble the appearance. Out-of-date aerial images or satellite data make a risky template. Another option is to use line templates from maps. These can be matched with contours extracted from the image. This is less appearance dependent, demanding no particular colours or grey-values. Careful path planning with manual choice of auspicious line sets from the map for each waypoint is required. The human navigator in his transparent cupola looked at the scene in a different manner.

Thus, automatic understanding of salient landmarks beneath the aircraft by cognitive approaches is of

interest. Preferred are objects with a clear, self-evident reference location – such as bridges, and crossroads where the intersection of the mid-axes makes a good reference. Also large representative or institutional buildings like churches or mosques are an option. They most often exhibit symmetric outlines, where the intersection of main symmetry axes constructs a good reference location. Less suited are objects with possibly repetitive structures such as large industrial plants. Though they may be quite salient in the perceptive sense, they provide no self-evident reference location and thus no clear set-position.

By automatic understanding we do not mean to mimic the whole cognitive reasoning of a human being. Instead we are more interested in (1) the *perceptive capabilities*, e.g., Gestalt grouping according to principles like symmetry, proximity, parallelism, repetition and similarity; (2) *part-of reasoning* that tends to see aggregates of simpler parts on many levels of scale; (3) *concretization*, i.e., knowledge on how certain simple 3D objects often appear in aerial images; and (4) simple *logical abductive inferences* following is-a hierarchies. The reader may imagine a navigator looking down and thinking "I see contours grouped nicely in good continuation; two of these run parallel forming a stripe; this stripe may well be a part of the road I am looking for; there is another such stripe; it runs parallel to yet a third one quite close; these two may well be the highway I am looking for; they end at the road-stripe maybe because of occlusion; that should be the bridge-over-highway that is supposed to be around here somewhere!" It is clear that such reasoning does not follow the rules of deductive logic. In particular, if the navigator knows that roads – like many other objects in a scene – appear as stripes, he may not infer logically that the stripe he perceives is a road, or even more specifically "the" road he is looking for. In fact the modelled road may even have no sufficient contrast to its surrounding and thus be invisible. Instead of producing plane truth, such reasoning is searching for plausible explanations of data, based on expectations or hypotheses. That does not mean that it cannot be performed by a machine using quite similar mechanisms as are appropriate for automatic theorem proving. Only the results have to be interpreted differently. Abductive reasoning produces plausible though uncertain explanations not plane truth. This paper discusses means to investigate the robustness of such knowledge-based landmark recognition.

1.2 Related Work

Automatic image understanding has a long history in particular for remote sensing applications [11]. Abduction as logic model next to approximate solving and plausibility is discussed in [12] for a system recognizing buildings using a layered model mimicking human vision. Sophisticated production systems have been proposed e.g. in [2, 6]. More contemporary work unifies statistical approaches with such syntactical or structural approaches [3]. It is difficult to achieve the low fault rates and high precision demanded by e.g., map update tasks in an open world where unknown objects will appear that have not been modelled yet. In a navigation control loop and fusion setting the requirements for error rates and precision can be set much lower without jeopardizing the overall robustness.

Some military UAVs and missiles already have automatic vision components included in their flight control. Emphasis is on real-time fusion of all available information sources – such as radar, GNSS, INS, star trackers, GIS-data, altimeters, and – at last – also vision [1]. Vision is regarded as valuable alternative drift-less source in case of GNSS failure [4].

Our own approach dates back more than 20 years [5, 13]. A recent renewal of this work assesses the structural approach for landmark-based UAV navigation, by closing a simulated control loop using Google Earth as camera simulator and image source [10]. There are two keys to swift any-time performance and robustness of such systems: 1) the inclusion of clustering or accumulating productions [7] and 2) top down search rationales in addition to the quality driven interpretation included in the interpretation mechanism [8]. The production system approach can also be used for different recognition purposes such as finding building outlines for GIS-update based on gestalt relations [9].

2.0 DECLARATIVE CODING USING PRODUCTION SYSTEMS

In this technical part of the work the proposed recognition method is explained in more detail.

2.1 Object Oriented Landmark Description

The regions of the world differ deeply in the kind of salient objects encountered there and in their frequency. An object-oriented system has to be flexible enough to load specific knowledge, e.g., from ontologies about the area, where the UAV will be operating in. This is more promising than the use of machine learning recognition techniques such as Support Vector Machines or statistical recognition based on image features which have been trained with non-representative data.

Object recognition classes of our system are inherited from CImageObject as can be seen from the two example class diagrams in Figure 1. For each class, that has no decomposition link, a constructor is required that can segment such object from the image – most often this is a filter operation followed by a threshold. We call these classes primitive. Here we only have one primitive CLine, which results from a gradient filter. So these are small contour segments.

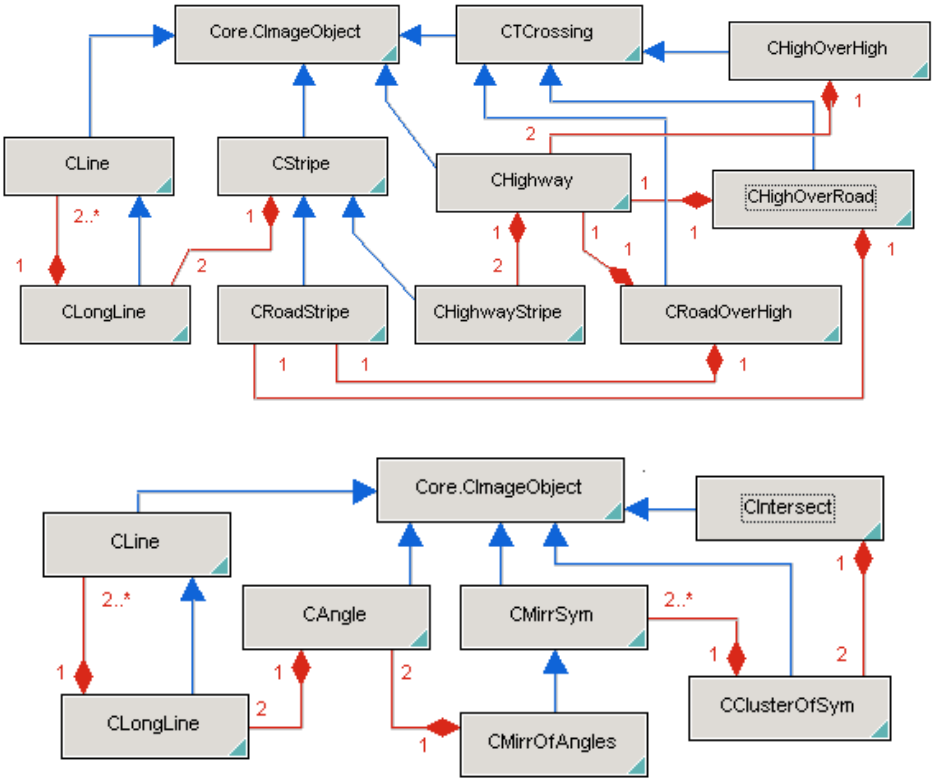

Figure 1: UML Class-diagrams displaying part-of links and is-a links

2.2 Instantiation as Search Administrating Hypotheses

The declarative description of parts and aggregates and concretizations as given above does not define a procedure of recognition. We may understand it as a particular kind of grammar. In [5] the BPI system is proposed as user-independent solution for accumulative interpretation of such production systems following the blackboard rationale. Such systems use a dispatcher assigning working hypotheses to computational resources. Such a hypothesis is called WorkingElement in Figure 2. It consists of a

triggering object instance (called ImageObject) and entries from a corresponding production rule (namely left-hand side, i.e. HypoType, partners in the right-hand side PartnerType – and, optionally, context). The dispatcher module gets the production system as input. Thus, if a WorkingElement has no hypothesis attached yet it will form admissible clones, else it will call the appropriate methods that test constraints, and if those hold new ImageObject instances will be produced. From each newly produced instance (and from primitives segmented from the input image) new WorkingElement instances are formed with no hypothesis attached yet.

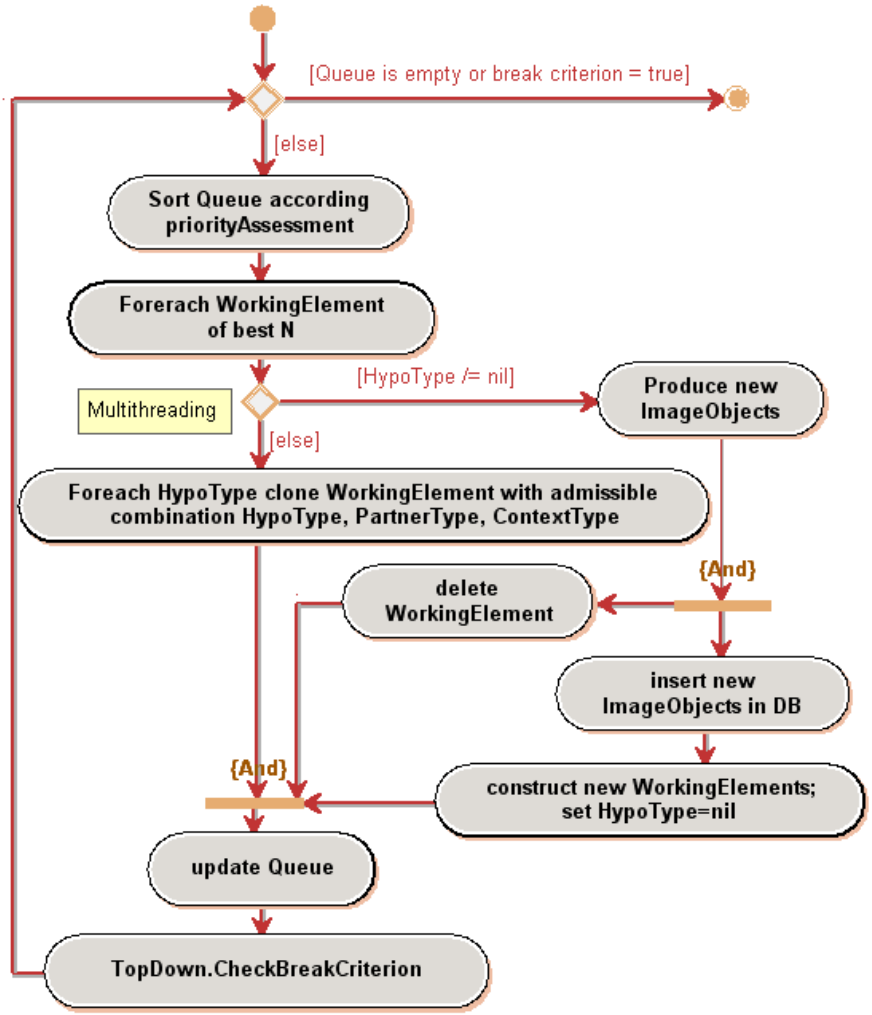

Figure 2: UML Activity-diagram for hypotheses administration

This cycle can be repeated until, either all hypotheses have been processed, or the object of interest has been instantiated, or other stop criteria (such as maximal admissible time) are meat. The set of WorkingElement instances is organized as Queue which is ordered according to an assessment value. Such value is by default given through a quality measure for the corresponding image object (data-driven search). Many systems have an additional assessment component – the importance. This is achieved by weight factors on the quality. Given a particular state of the search WorkingElement instances gain different importance for the task at hand – particular HypoTypes will be of more interest, instances in particular image regions may be of higher or lower relevance. Such use of top-down importance for

focusing the search is described in detail in [8]. Both assessment components (quality and importance) have to be provided by the user.

3.0 VIRTUAL GLOBE SYSTEMS AS TEST-BED

Any recognition system has to be assessed with respect to the requirements of the task it is intended for. Accordingly, for vision based UAV navigation the gold-standard would be flying a real vehicle over the intended terrain and counting how often it runs astray. Since this may currently be hazardous, prohibited or quite expensive it should be simulated in an appropriate way. Internet-based virtual globe systems (VGS) such as Google Earth provide an almost open world and a camera simulator which yields a picture for any given geo-coordinate. This provides a very valuable data source which was not available thirty years ago when we first proposed such production systems.

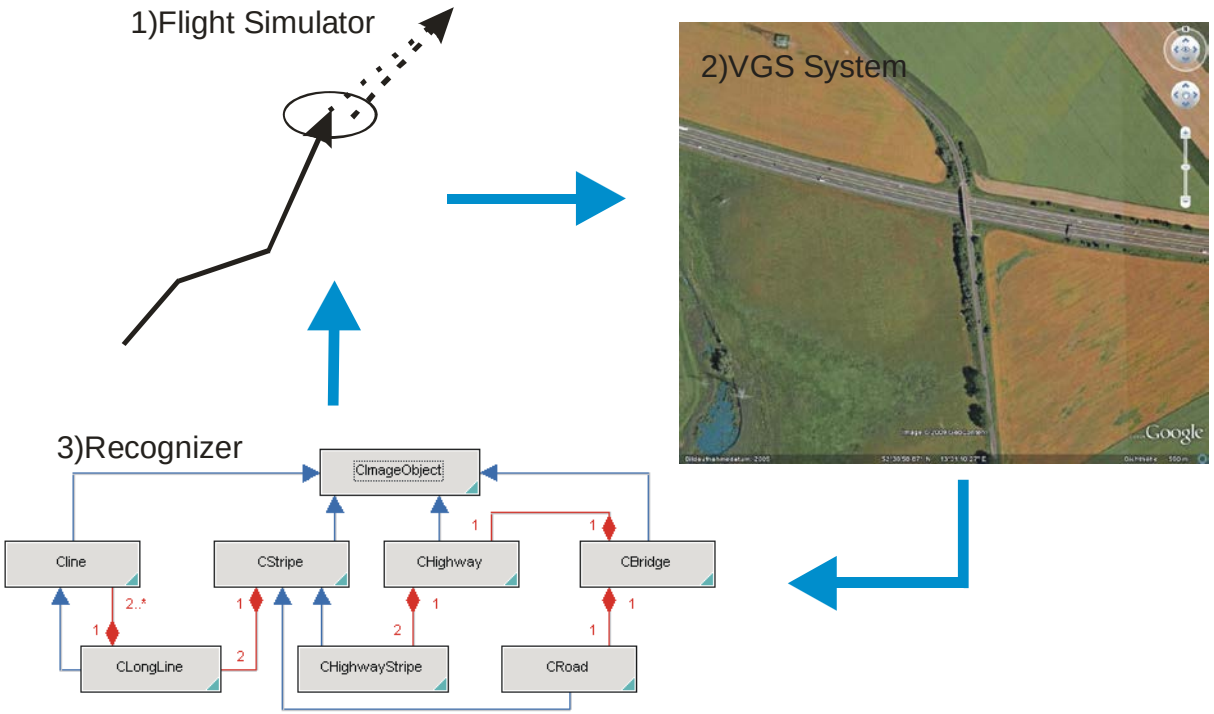

Figure 3: Simulated perception-action loop featuring knowledge-based recognition and VGS

Figure 3 shows the simulation principle: The *set-flight-path* is given as polygon of *set-points*. Each such set point is attributed with a geo-coordinate. At all these locations there is a salient landmark object. The flight-navigation simulation system constructs successively the *current-flight-path* by following the *set-flight-path* and adding a drift obtained from a random generator. Each time the current-position is transferred to the VGS system which takes a picture there. This picture is fed into the recognition system. After a certain amount of time the search for plausible landmarks is terminated. A decision is made for the most plausible location in the image. This is given again to the flight-navigation simulation system which subtracts the deviation from the *current-flight-path* thus correcting – hopefully – the drift error. With this closed loop automatically very many experiments located all over the world can be made (and repeated) limited only by the available computational resources and time. The robustness of such navigation approach can be assessed properly. In the following subsections each part of the test-bed is described in more detail.

3.1 Setting a Flight-path on the map-layer of a VGS

In particular if the path is set by one of the recognition system developers he should not see the pictures that later in the experiments will be fed into it. Otherwise the results may be biased in favour of the system. It is recommended to use the map-layer (which usually is also provided with VGS systems).

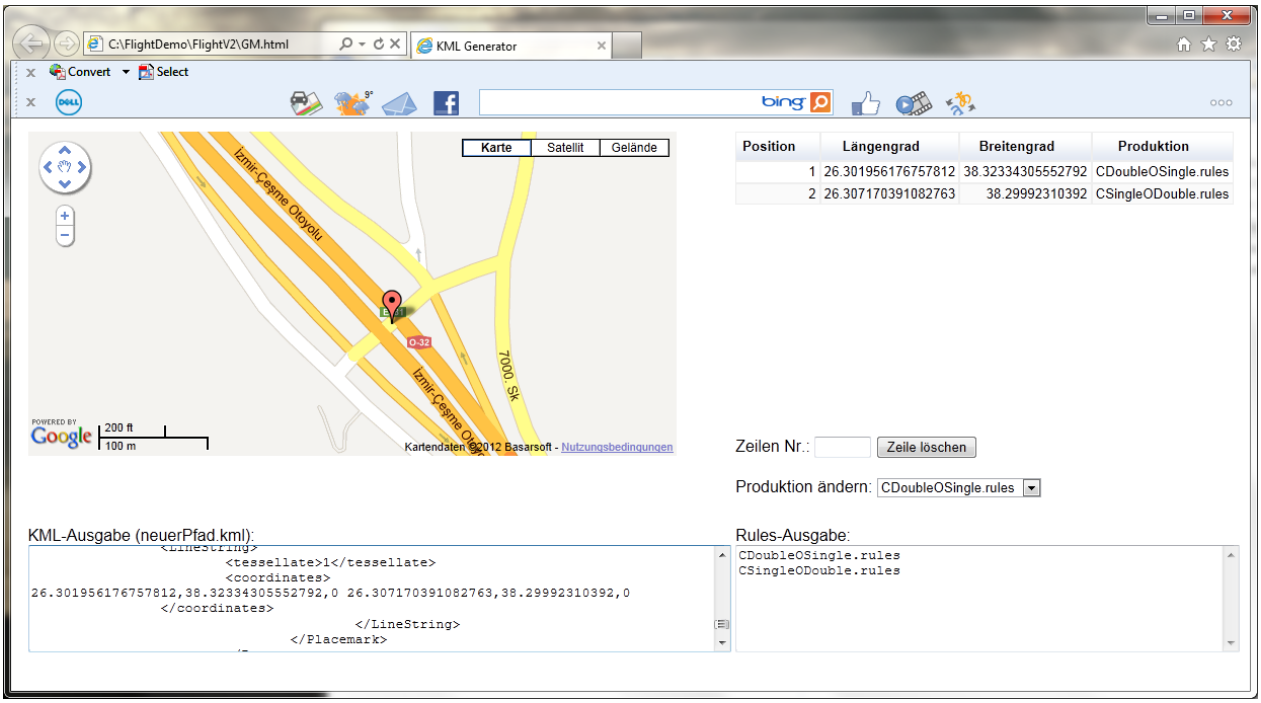

Figure 4: Screen-shot of HTML-script interface for set-path definition

Figure 4 shows the interface usually used for defining a new *set-flight-path* using a HTML-script. This script calls the map-layer of the Google VGS. The second set-point of a path in Western Turkey is set on a bridge leading the Izmir-Cesme highway over a secondary road. Note that the flight height is set to zero in the KML file (KML is a XML variant for communicating such paths).

3.2 Flight-navigation simulation

Planet earth is often modelled using ellipsoid coordinates, where different organisations (and VGS providers) recommend and use different ellipsoids. For simplicity this work uses a simple spherical model and for each flight only one tangent space. Thus the latitude-angle is transformed into meters-north in a plane using the mid Earth-radius; and the longitude-angle is transformed into meters using the cosine of the latitude of the first set-point as additional global factor for the whole path. By this simplification all navigation is performed in a metric 2D plane. On the other hand, for paths going very far in South or North direction, we thus accept a considerable distortion due to the curvature of the planet. Our longest paths in Germany had e.g. about 600km size in that direction, but we could not observe any influence from this. Of course for experiments with longer paths a more appropriate mapping should be included.

The flight altitude is fixed at 660m and the relief modelling of the VGS disabled. Depending on the size of the VGS-window on the screen this yields a ground sampling distance of 0.7m for a pixel. This factor has to be roughly calibrated in advance. The screenshots of the VGS-window are larger than 512×512 pixels so that the images can be cropped out of the centre – in order to avoid influence from the logos.

Different INS-drift models can be included. Here a simple Gaussian drift error is used where bias and standard deviation are growing linearly with the path length since the last update. Our default setting is standard deviation 2‰ of the last path segment and a bias of 1‰ east. Thus a blind reckoning flight over 200km distance is expected to be 200m off target with a circular 2σ disk of 1600m diameter. So the target would probably not be visible in an image taken there. But, if the distance between set-points is on average some 10km not even all landmarks must be found. The navigation can stay along the track even if two or three in a row are not found. It will, however, go astray if the navigation decides for wrong landmark detections far off the image centre.

3.3 Decision

The result of a recognition run is a set of more or less plausible landmark positions. Mutual consistent detections are clustered. Clusters farther away from each other are of course mutually contradicting. A navigation control loop needs a rationale on how to decide for one of these possibilities. Here we distinguish three different types of such decision:

- *Optimistic:* The most dominant and thus most plausible cluster of consistent detections of a landmark is accepted as correct. This is a natural decision from the perspective of artificial intelligence because it follows the inferences the machine has done. Accordingly, the current drift is estimated to equal the difference vector between this clusters location and the image centre, transformed to world coordinates. This is used as course correction.

- *Pessimistic:* This rationale has no confidence in the machine-vision system at all. It just continues dead reckoning. Thus a growing drift from the set-points is to be expected – see Figure 5 bottom left.

- *Heuristic:* Common sense teaches that a detection should not be trusted if it is too far away from the image centre. At least one extra parameter weighting the distance from the image centre against the plausibility of the detection clusters is required. Knowing the combinatorics of the system, the sixth root of the number of detections in a cluster is set against the squared distance from the centre. The weight factor is chosen heuristically. This needs experience and knowledge – but no statistic.

4.0 EXPERIMENTS AND DISCUSSION

Set-up, debugging and parameterization of the system have been done using few paths over Germany along the highways. Mostly heuristic decision rationale was applied. Usually, the system will run satisfactory over rural areas; in particular if the highway's concrete surface is old and thus bright. There are problems with 1) winter pictures (e.g., with stripes of snow that happen to fit the correct width); 2) urban, sub-urban or industrial regions (e.g., roofs may be mistaken for highway or road parts); 3) very new road surface sometimes giving a similar grey-tone as the surrounding so that contours are weak. Figure 5 displays some typical error records.

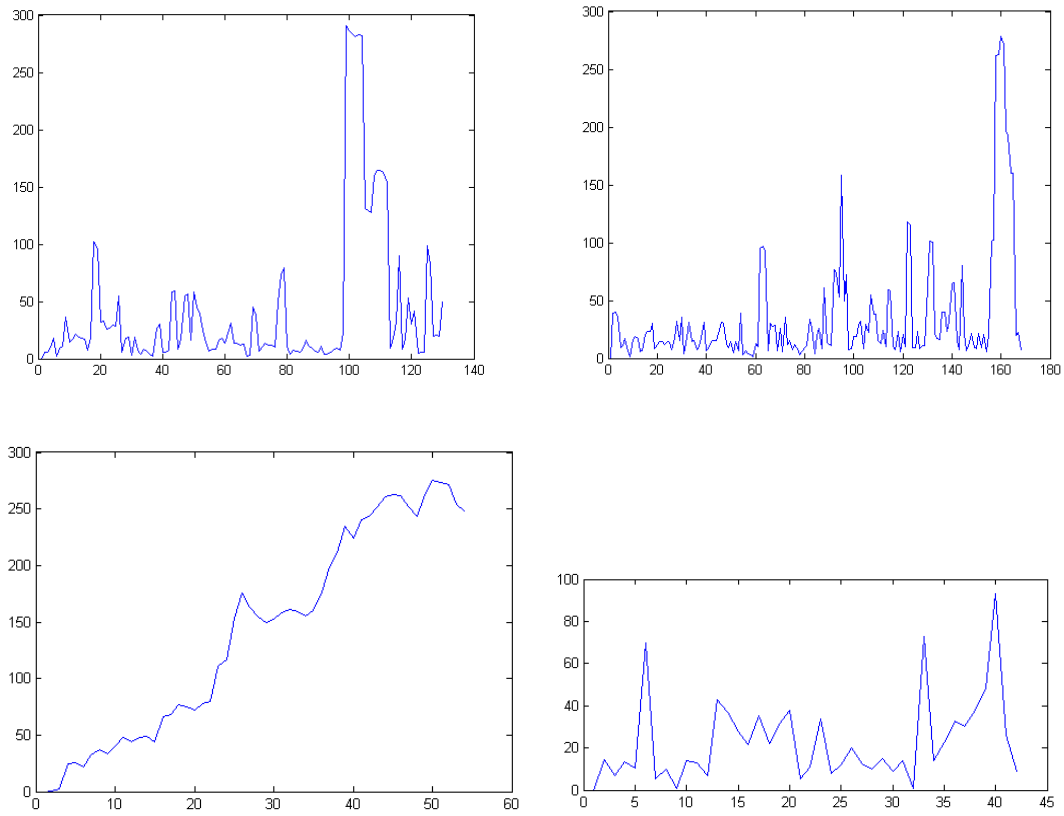

Figure 5: Records of flight-errors (distance between *set-point* and *current-point* in m): Top row, examples with recovering from gross recognition error; lower left blind dead-reckoning example; lower right successful standard example

Frequently, deviations of up to about 50m occur. It is observed that such behaviour results from mistaking structures close to and parallel to the highway for the real thing. But the flight usually returns to the *set-path* again. Gross errors that intermittingly lead the flight astray also occur now and then. But the system exhibits a remarkable capability to recover from such gross mistakes. Compared to the pessimistic rationale, flying with the heuristic or even optimistic rationale gives a much higher chance to reach the target. A detailed quantitative analysis is under way and to be published. This will then also give way to decide according to the Bayes-rule using an empirically acquired likelihood – adding a forth rationale: The *optimal* decision.

In conclusion a structural knowledge-based landmark navigation system may help providing absolute positions when GNSS fails. It may well be adapted to different geographical scenarios provided that the system is capable of easily including new knowledge in the form of new classes suiting the new scenario. It is evident that in populated regions where landmarks of known structure and measures such as major highways are missing large salient buildings formed according to known common principles – such as mosques or churches – can be used as persistent geographical landmarks. While a learning system would require a new representative data set, a knowledge-based system requires new scene specific descriptive knowledge. Since this evidence is from a complete different source – its fusion with the other known and used sources is promising.

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.* In: Z. Rahman, S. E. Reichenbach, M. A. Neifeld: (Eds.) *2007 SPIE XVI*, Vol. 6575. 8-13, April 2007.

[2] B. Draper, R. Collins, J. Brolio, A. Hanson, E. Riseman: *The Schema System*. IJCV, (2), 1989, pp. 209-250.

[3] C.-E. Guo, S. C. Zhu, Y. N. Wu: *Modelling visual patterns by integrating descriptive and generative methods*. International Journal on Computer Vision, 53 (1): 5-29, 2003.

[4] Z. He, R. V. Iyer, P. Chandler: *Vision-based UAV Flight Control and Obstacle Avoidance*. IEEE American Control Conference, Minneapolis, MN, 2166 –2170, June 2006.

[5] K. Lütjen: *Ein Blackboard-basiertes Produktionssystem für die automatische Bildauswertung*. In: G. Hatmann (ed.): *Mustererkennung 1986*, DAGM 1986. Informatik Fachberichte 125, Springer, Berlin: 164-168, 1986.

[6] T. Matsuyama, V.S.-S. Hwang: *Sigma a Knowledge-based Image Understanding System*. Plenum Press, New York. 1990.

[7] E. Michaelsen, L. Doktorski, M. Arens: *Shortcuts in Production Systems – A Way to Include Clustering in Structural Pattern Recognition*. PRIA-9 2008, vol. 2. : 30–38, Lobachevsky State University, Nizhni Novgorod, ISBN: 978-5-902390-14-5, 2008.

[8] E. Michaelsen, M. Arens, L. Doktorski: *Interaction of Control and Knowledge in a Structural Recognition System*. In: Mertsching, B., Hund, M., Aziz, M. Z. (eds.) *KI 2009*, LNAI 5803, Springer, Berlin 2009, pp. 73-90.

[9] E. Michaelsen, U. Stilla, U. Soergel, L. Doktorski: *Extraction of building polygons from SAR images: Grouping and decision-level in the GESTALT system*. Pattern Recognition Letters, to appear 2010, online under http://dx.doi.org/10.1016/j.patrec.2009.10.004

[10] E. Michaelsen, K. Jaeger. *A GOOGLE-Earth Based Test Bed for Structural Image-based UAV Navigation*. 12th International Conference on Information Fusion, Seattle, WA, USA, July 6-9, 2009, IEEE-ISIF, Proc. on CD, ISBN: 978-0-9824438-0-4, pp. 340-346.

[11] M. Nagao, T. Matsuyama. *A Structural Analysis of complex Aerial Photographs*. Plenum Press, New York, 1980.

[12] T. Schenk: *A Layered Abduction Model of Building Recognition*. In: A. Gruen, O. Kuebler, P. Agouris (eds): *Automatic Extraction of Man-made Objects from Aerial and Space Images*. Birkhäuser, Basel, 1995, pp. 117- 123.

[13] U. Stilla and E. Michaelsen: *Semantic Modelling of Man-made Objects by Production Nets*. In: A. Gruen, E. P. Baltsavias, O. Henricsson (eds.): *Automatic Extraction of Man-made Objects from Aerial and Space Images (II)*. Birkhaeuser, Basel, 1997, pp. 43-52.