

AIMS: An SQL-based System for Airspace Monitoring

Gereon Schüller, Andreas Behrend

{gereon.schueller@fkie.fraunhofer.de, behrend@cs.uni-bonn.de}

Fraunhofer FKIE, Dept. Sensor Data Fusion

Neuenahrer Straße 20, D-53343 Wachtberg, Germany

Abstract: The “Airspace Monitoring System” (AIMS) is a system for monitoring and analyzing flight data streams with respect to the occurrence of arbitrary complex events. It is a general system that allows for a comprehensive analysis of aircraft movements, in contrast to already existing tools which focus on a single task like flight delay detection. For instance, the system is able to detect critical deviations from the current flight plan, abnormal approach parameters of landing flights as well as areas with an increased risk of collisions. To this end, tracks are extracted from cluttered radar data and SQL views are employed for a timely processing of these tracks. Additionally, the data is stored for later analysis.

1 Tracking Flights

Airspace monitoring systems visualize the current state of aircrafts based on radar observation or on transponder signals, or both. However, there are still plenty of situations where highly critical events are detected too late, like the Ueberlingen catastrophe has shown [4] that occurred due to the inattentiveness of the responsible flight controller. There are many critical events in airspace, like close encounters or violations of no-fly zones. Disappearance of flights from their scheduled route are another anomaly of interest; they can indicate hijacking of airplanes. The development of automated tools able to detect critical events based on streams of continuously arriving sensor data and to raise an alarm triggering human (or automatic) action is therefore a highly relevant, but still mostly unsolved task. First of all, it is necessary to track the airplanes. A *flight track* is a sequence of probability density functions representing an estimation of a flight path. It is derived from measurements either by primary radar or by transponder signals of equipped aircrafts. But the raw position measurements suffer from several drawbacks: Radar or transponder signals may be missed due to physical effects, and false detections may arise from punctual disturbing sources like clouds, birds etc. Misdetetection and false alerts/clutter are connected anti-proportionally as leveraging the amplification of the signal will also amplify the disturbing signals [5]. Another problem consists in the fact that radar signals are not sharp in position and will be smeared out. If multiple objects are in the field of view, the assignment between plot and corresponding plane has to be solved.

All these effects make the process of tracking complex. Many tracking algorithms have been proposed in the past decade, e.g. [6, 7, 8]. The development of new and enhanced tracking algorithms is still an active research area, in particular at Fraunhofer FKIE. Track-

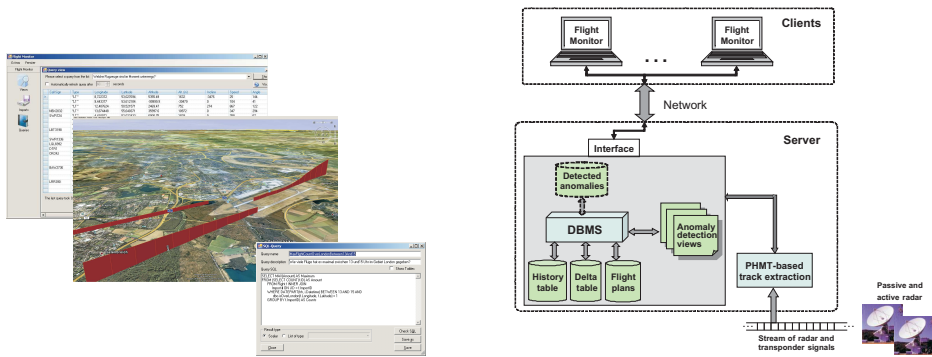


Figure 1: GUI of the Flight Monitor (left), Architecture of AIMS (right)

ing algorithms can be distinguished by several characteristics. For example, there are trackers that can cope with multiple targets and will compute several hypotheses about which plot belongs to which target. Other trackers feature prediction, estimating how the target will behave in future, or they may feature retrodiction, calculating what we can say about the way in the past by knowing the current results. In our approach, a Bayesian tracking algorithm is employed. It is important to know that trackers usually use stochastic models and will output the kinematic state together with a quality indicator of the estimation, i.e., the covariance in position and speed as well as a measure for the size of the region in which the object can be estimated with a certain probability. Thus, a sample output of our tracking algorithm may look as follows:

timestamp	id	lon	lat	height	covX	covY	vel	heading
19:20:43	34	51.124	7.054	4534	30.2	1.2	300.4	91
19:20:44	35	50.985	6.345	2324	40.4	1.4	240.3	27
...

2 The AIMS System

AIMS is a prototype of a system for monitoring and analyzing flight movements. It has been developed at the University of Bonn in cooperation with Fraunhofer FKIE and EADS Deutschland GmbH. The aim is to develop efficient DBMS-based methods for real-time gathering and monitoring of streams of flight data. Currently, the system is used to monitor the complete German airspace every 4 seconds with up to 2000 flights in peak times. An overview of the AIMS architecture is given in Figure 1.

Our system basically consists of three components. The first one is a *track extraction* tool used for detecting and identifying aircrafts using radar data and transponder signals. Since radar data usually contain noise, we employ the *probabilistic multiple hypothesis*

tracking (PMHT) algorithm for detecting moving objects in a cluttered environment [8]. The PMHT algorithm calculates a sequential likelihood ratio for solving the plot-to-track association problem, testing the hypothesis that a plot belongs to a track. The resulting tracks and the transponder signals are then merged into a stream of highly accurate aircraft positions. The resulting stream of timestamped position and velocity data (track data for short in the following) is periodically pushed into a relational database. For programming this database application, we used Microsoft SQL Server because of its well-optimized user-defined function (UDF) processing. UDFs are, e.g., used for spatial coordinate transformations and vector computations. In our scenario, new track data is provided every four seconds and stored in a 'delta table' containing just the most recent track data. Its former content is moved to the history table such that the complete course of each flight is recorded. However, before data is stored in the database, a lot of data transformation and cleansing has to be done for which we employ SQL triggers.

The main feature of our approach to stream monitoring is to use continuous SQL queries (stored as views) for specifying anomalous situations to be detected, as well as important application-specific concepts and parameters required for anomaly detection. A similar approach has been already used in various other successful applications by our group ([1, 2]). In the flight tracking scenario, the following questions are to be answered by the AIMS system:

1. Which aircrafts are currently airborne?
2. Which aircrafts are currently landing?
3. Are there aircrafts on collision course?
4. Are there critical deviations from flight plans?
5. How many aircrafts are currently over a certain region?
6. What is the average number of landings for an arbitrarily chosen airport?

The view definitions can be created in the so-called *Flight Monitor*, a graphical user interface written in Visual Basic and C# that allows the user to interact freely with the system (see Fig. 1). Its main tasks are the support for defining new anomaly detection views (although the resulting views are then stored in and managed by the relational database system) and the continuous textual and graphical monitoring of their results. The incorporated SQL editor allows to freely define new detection views which may directly access the underlying tables and/or other anomaly detection views already defined. In this way, various conditions of given detection views can be combined in order to define complex events with respect to critical aircraft movements.

In addition, the Flight Monitor conducts performance measurements for each periodic query execution. After considerable tuning, all of the above queries could be executed in less than 2 seconds (except for the last one determining average values), being below the refreshment rate of 4 seconds. Another feature of the Flight Monitor is its graphical visualization of the detected anomalies by means of an exported KML (keyhole markup language) file which can be processed by several programs. In our case, Google Earth is employed as long as our system remains in a prototypical status. In order to achieve a meaningful graphical visualization, the anomaly detection views usually retain attributes for position and time values of the flight data under consideration.

3 Incremental Materialization for Performance Enhancement

Even though a considerable degree of analysis can be achieved by purely recomputing expressive SQL views in each refreshment cycle, the given stream scenario will sooner or later drastically slow down our system without further optimizations.

To this end, we use an incremental approach to materialize the running queries. The track data are separated into two relations; one for the data of the current sliding window, the other for older track data. While most queries will only run on the newest data, some queries will include the usage of historic track data, but only those who are related to the newest data, allowing for the usage of joins between new and old data. Index structures are used to support fast access to these historical data. The update propagation process, as well as moving data into the relation for historical data is done by a system of cascading triggers that fire on the input of arriving data. This way, the views are always up to date after insertions. Some queries do not fall into these categories, using large portions of historical track data, e. g. "What is the average number of landings for an arbitrarily chosen airport?". However, the nature of these queries is not a running one, they are merely use for statistics and are executed infrequently.

First evaluation results already indicate that conventional SQL queries can be used for efficiently processing this realistic stream scenario [3]. Since AIMS is still in a prototypical state, however, a comprehensive performance evaluation cannot be presented yet. However, all presented anomaly detection views - which include the determination of landing and departing flights, critical approaches, deviations from flight plans as well as the determination of delay times - could be executed in less than 2 seconds. AIMS provides a more flexible approach to airspace monitoring allowing the free definition of arbitrary complex events over a stream of flight data. The flexibility results from using SQL views which freely add and combine user-defined anomaly detection view specifications.

References

- [1] A. BEHREND, C. DORAU, R. MANTHEY, AND G. SCHÜLLER: *Incremental view-based Analysis of Stock Market Data Streams*. IDEAS 2008: 269-275.
- [2] A. BEHREND, R. MANTHEY, G. SCHÜLLER, AND MONIKA WIENEKE: *Detecting Moving Objects in Noisy Radar Data Using a Relational Database*. ADBIS 2009: 286-300
- [3] G. SCHÜLLER, A. BEHREND, AND R. MANTHEY: *AIMS: An SQL-Based System for Airspace Monitoring*. IWGS 2010: 31-38
- [4] BUNDESSTELLE FÜR FLUGUNFALLUNTERSUCHUNG: *Investigation Report AX001-1-2/02*, May 2004.
- [5] M. I. SKOLNIK: *Introduction to Radar Systems*. McGraw-Hill, 1981.
- [6] S. S. BLACKMANN AND R. POPOLI: *Design and Analysis of Modern Tracking Systems*. Artech House, Boston, USA, 1999.
- [7] G. VAN KEUK. Mht Extraction and Track Maintenance of a Target Formation. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(1):288–295, Jan 2002.
- [8] M. WIENEKE, W. KOCH: On Sequential Track Extraction within the PMHT Framework. *EURASIP Journal on Advances in Signal Processing*, 2008: 1–13.