ABSTRACT

In previous work, we have proposed a tool for Spatiotemporal Pattern Query. It matches individual moving object trajectories against a given movement pattern. For example, it can be used to find the situations of Missed Approach in ATC data (Air Traffic Control systems, used for tracking the movement of aircrafts), where the landing of the aircraft was interrupted for some reason. This tool expresses the pattern as a set of predicates that must be fulfilled in a certain temporal order. It is implemented as a Plugin to the Secondo DBMS system. Although the tool is generic and flexible, domain expertise is required to formulate and tune queries. The user has to decide the set of predicates, their arguments, and the temporal constraints that best describe the pattern. This paper demonstrates a novel solution where a Visual Analytics system, V-Analytics, is used in integration with this query tool to help a human analyst explore such patterns. The demonstration is based on a real ATC data set.

Categories and Subject Descriptors
H.2.4 [Systems]: Query processing—Secondo, V-Analytics; H.3.3 [Information Search and Retrieval]: Query formulation, Information filtering, Selection process.

General Terms
Experimentation, Languages.

Keywords
Secondo, V-Analytics, Spatiotemporal Patterns.

1. INTRODUCTION

This paper demonstrates the use of the visual analytics system V-Analytics [4] in integration with the moving objects DBMS Secondo [2] to explore Spatiotemporal Patterns (STP) to analyze a real world data set. Secondo brings an extensive set of query operations accessible through a query language to let the user express arbitrarily complex queries and efficiently evaluate them on large moving objects databases. On the other hand, V-Analytics enables the user to pre-process the data, select an interesting subset for analysis, tune parameters of the sophisticated queries in Secondo and interpret their results.

In this paper, we demonstrate two pattern exploration tasks on a data set that includes one day (Feb 22, 2008) radar recording of aircraft positions (Lon, Lat, Alt) over France and neighboring territories, consisting of 17,851 trajectories and 427,651 recorded positions. The sampling interval is variable, ranging from about one to about three minutes. The data has been anonymized by removing aircraft identities and airline information. In this data set, we search for the following landing patterns:

(1) Missed Approach: while the aircraft is in its final approach (i.e., the landing phase) the pilot for some reason decides to climb again. In order to improve the safety, it is worthwhile to analyze such hazardous situations. Such analysis can highlight redundant situations that can be solved by changing airways, approach procedures, etc.

(2) Stepwise Descent Landing: a landing pattern in which the aircraft alternates between descent and cruise till it touches the ground. This is a less desirable landing pattern compared to the Diving pattern, where the aircraft constantly descends till it lands. The latter is preferable because of lower fuel consumption and lower pollution and noise.

Studying such patterns in the ATC data helps to improve the air traffic. The rest of this paper is organized as follows.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGSPATIAL GIS ’11, November 1-4, 2011, Chicago, IL, USA.
Section 2 briefly describes the two systems Secondo and V-Analytics, and discusses issues related to their integration. Section 3 briefly describes the Spatiotemporal Pattern Predicate. Section 4 illustrates what will be demonstrated. Finally we conclude in Section 5.

2. INTEGRATING SECONDQ AND V-ANALYTICS

Secondo [2] is an open source extensible DBMS platform. It consists of three modules: the kernel, the GUI and the query optimizer. It contains an extensible set of algebras, each of which is defining database types and query operations. It contains algebras implementing a large part of the moving objects database model in [7]. We provide the implementation of the spatiotemporal pattern predicate as a Secondo Plugin [1].

V-Analytics [4] incorporates various visualization techniques, interactive tools, and computational methods for analyzing spatial, temporal, and spatiotemporal data. Among them are time-aware maps, space-time cubes, other types of graphs and diagrams; interactive dynamic filtering of data according to their spatial, temporal, and thematic (attributive) components; computational procedures oriented to movement data such as clustering of trajectories [5], generalization and summarization [3], extracting various types of events from movement data, etc.

Secondo and V-Analytics are integrated so that it is possible to exchange data between the two systems in both directions. The moving object trajectories are initially loaded into the databases of the two systems. One can start by visualizing the data in V-Analytics, select a set of candidate trajectories, and/or interesting time periods within their life times, then send the trajectory identifiers and periods to Secondo for further processing. In the other direction, one can perform sophisticated query operations in Secondo (e.g., STP queries) and compute derived attributes, then transfer the query results to V-Analytics for visualization and further analysis. This process of transferring data between the two systems allows the user to explore the patterns in the data effectively.

3. THE SPATIOTEMPORAL PATTERN PREDICATE

Consider the Missed Approach procedure in the Section 1. It can be described by three predicates: aircraft comes close to destination, aircraft descends till it is below a certain altitude, aircraft climbs up again. Temporally, the third predicate must be fulfilled after the second predicate, and both of them must be fulfilled during the fulfillment time of the first predicate.

The STP Predicate is proposed in [8]. It is based on the abstract data types model for moving objects in [7] [6]. The model defines the moving type constructor. It constructs the time-dependent counterpart of every static type. The moving types are denoted by prefixing m to the standard types (e.g., mpoint). We use the italic underline style all over the paper to denote types. The mpoint, for example, is represented as a temporally ordered list of units (mpoint), each of which consists of a time interval and a line function. The coordinates of the mpoint at any time instant within the interval are obtained by evaluating the line function.

The STP predicate is a pair (P, C), where P is a set of time-dependent predicates, and C is a set of temporal constraints on the fulfillment times of the predicates in P. Time-dependent predicates were first introduced in [7]. Figure 1 illustrates an example. The Trip is an mpoint object, that crosses the region object R between the time instants 378.3 and 397.7. The evaluation of the time-dependent predicate Trip inside R is a time-dependent boolean (mbool) having the value true between these two time instants and false otherwise. A time-dependent predicate, hence, is an operation that yields an mbool. All the static predicates defined for non-moving types can uniformly and consistently be made applicable to the corresponding moving types as time-dependent predicates, see [7].

The temporal constraints specify the order of fulfillment of the time-dependent predicates. The Missed Approach procedure can be expressed as follows:

\[ \ldots \text{stpattern} \]
\[ \text{Close}: \text{distance}(.\text{Position}, .\text{Destination}) < 5000.0, \]
\[ \text{Down}: ((.\text{AltitudeDerivative} < 0.0) \text{ and } (
\text{.Altitude} < 1000.0)), \]
\[ \text{Up}: .\text{AltitudeDerivative} > 0.0; \]
\[ \text{stconstraint}("\text{Close}, "\text{Down}, \text{vec}(\text{"abba"}, \text{"a.bba"}, \text{"baba"}), \)
\[ \text{stconstraint}("\text{Close}, "\text{Up}, \text{vec}(\text{"abba"}, \text{"aba.b"}, \text{"abab"}), \)
\[ \text{stconstraint}("\text{Down}, "\text{Up}, \text{vec}(\text{"aabb"}, \text{"aa.bb"})) \ldots \]

Here we show only the STP predicate (expressed in Secondo using the stpattern operator) and omit the rest of the query due to space limitation. We assume that it gets a tuple of type:
\[ \text{tuple}[\text{Position}, \text{mpoint}, \text{Destination}; \text{point}, \text{Altitude}; \text{mreal}, \text{AltitudeDerivative}; \text{mreal}], \]
where Position is the time-dependent (Lon, Lat) of the flight. The stpattern operator accepts two argument lists, separated by a semicolon. The first list contains three time-dependent predicates having the aliases Close, Down, and Up. The second list contains three temporal constraints, expressed by the stconstraint operator, each of which specifies a temporal relation between two time-dependent predicates.

Each of the terms/arguments of the vcc operator specifies a relation between two time intervals. The start and the end time instants of the first interval are denoted \( s_0 \) and those of the second interval are denoted \( s_1 \). The order of the symbols describes the interval relationship visually. The dot symbol denotes the equality. For example, the relation \( a_0.b.b \) between the intervals \( i_1, i_2 \) denotes the order: \( (i_1.t_1 < i_2.t_2) \land (i_1.t_2 = i_2.t_1) \land (i_2.t_1 < i_2.t_2) \). The temporal relation expressed by the vcc operator is the disjunction of its components. Now we generalize this relation between two time intervals into a relation between two mbool objects.

Given \( P = \{ p_1, \ldots, p_n \} \) a set of time-dependent predicates, \( C = \{ c_1, \ldots, c_n \} \) a set of temporal constraints, and a tuple \( u \). Let \( p_i(u) \) denote the evaluation of \( p_i \) for the tuple \( u \) (i.e.,
Within the following section. Figure 2 illustrates one query result. It shows the altitude curve and the 2D trajectory projection of an aircraft that performed a Missed Approach procedure at the Châteauroux-Déols airport. The green color highlights the Down1 and Down2 events, and blue highlights Up.

For the given data set, the runtime of this query ranges from few seconds to half a minute depending on the amount of selected trajectories. The same holds for the query in the following section. Figure 2 illustrates one query result. It shows the altitude curve and the 2D trajectory projection of an aircraft that performed a Missed Approach procedure at the Châteauroux-Déols airport. The green color highlights the Down1 and Down2 events, and blue highlights Up.

4. WHAT WILL BE DEMONSTRATED

In this Demo we explore the Missed Approach and the Stepwise Descent Landing procedures, described in Section 1, using the proposed integration scheme. The typical use scenario for exploring patterns is: (1) in V-Analytics, select a subset of candidate trajectories, and transfer their ids to Secondo, (2) in Secondo, issue queries to locate the pattern instances (i.e., using the variants of the STP predicate), and to compute additional attributes, (3) transfer the query results to V-Analytics for visualization and analysis, and (4) if necessary, adjust the parameters and return to steps (1), (2) and (3).

4.1 Missed Approach

In this exploration task, we started in V-Analytics and selected the trajectories that landed at the end of their observed trips, to filter out noisy and incomplete data and to eliminate transit flights. In Secondo, we received the identifiers of this subset, and issued the query in Section 3 for Missed Approach. After several cycles of adjusting the parameters, the final query that we have reached is:

\[
\begin{align*}
\text{stconstraint}(&"Close", "Up", \text{vec}(\"abba\",\"aba.b\",\"abab\")), \\
\text{stconstraint}(&"Down1", "Up", \text{vec}(\"aabb\",\"aa.bb\")), \\
\text{stconstraint}(&"Up", "Down2", \text{vec}(\"aabb\",\"aa.bb\")); \\
\text{is_within_total_turn_range}.(\text{Pos at periods theRange}(end("Up"), inst(final(.Position)), TRUE, FALSE), 50.0, 270.0) \\
\text{filter}[\text{isdefined}(\text{Close})] \ldots
\end{align*}
\]

to eliminate transit flights. In served trips, to filter out noisy and incomplete data and selected the trajectories that landed at the end of their observation. The integration of selected trajectories. The same holds for the query in the following section. Figure 2 illustrates one query result. It shows the altitude curve and the 2D trajectory projection of an aircraft that performed a Missed Approach procedure at the Châteauroux-Déols airport. The green color highlights the Down1 and Down2 events, and blue highlights Up.

4. WHAT WILL BE DEMONSTRATED

In this Demo we explore the Missed Approach and the Stepwise Descent Landing procedures, described in Section 1, using the proposed integration scheme. The typical use scenario for exploring patterns is: (1) in V-Analytics, select a subset of candidate trajectories, and transfer their ids to Secondo, (2) in Secondo, issue queries to locate the pattern instances (i.e., using the variants of the STP predicate), and to compute additional attributes, (3) transfer the query results to V-Analytics for visualization and analysis, and (4) if necessary, adjust the parameters and return to steps (1), (2) and (3).

4.1 Missed Approach

In this exploration task, we started in V-Analytics and selected the trajectories that landed at the end of their observed trips, to filter out noisy and incomplete data and to eliminate transit flights. In Secondo, we received the identifiers of this subset, and issued the query in Section 3 for Missed Approach. After several cycles of adjusting the parameters, the final query that we have reached is:

\[
\begin{align*}
\text{stconstraint}(&"Close", "Up", \text{vec}(\"abba\",\"aba.b\",\"abab\")), \\
\text{stconstraint}(&"Down1", "Up", \text{vec}(\"aabb\",\"aa.bb\")), \\
\text{stconstraint}(&"Up", "Down2", \text{vec}(\"aabb\",\"aa.bb\")); \\
\text{is_within_total_turn_range}.(\text{Pos at periods theRange}(end("Up"), inst(final(.Position)), TRUE, FALSE), 50.0, 270.0) \\
\text{filter}[\text{isdefined}(\text{Close})] \ldots
\end{align*}
\]

to eliminate transit flights. In served trips, to filter out noisy and incomplete data and selected the trajectories that landed at the end of their observation. The integration of selected trajectories. The same holds for the query in the following section. Figure 2 illustrates one query result. It shows the altitude curve and the 2D trajectory projection of an aircraft that performed a Missed Approach procedure at the Châteauroux-Déols airport. The green color highlights the Down1 and Down2 events, and blue highlights Up.

4. WHAT WILL BE DEMONSTRATED

In this Demo we explore the Missed Approach and the Stepwise Descent Landing procedures, described in Section 1, using the proposed integration scheme. The typical use scenario for exploring patterns is: (1) in V-Analytics, select a subset of candidate trajectories, and transfer their ids to Secondo, (2) in Secondo, issue queries to locate the pattern instances (i.e., using the variants of the STP predicate), and to compute additional attributes, (3) transfer the query results to V-Analytics for visualization and analysis, and (4) if necessary, adjust the parameters and return to steps (1), (2) and (3).

4.1 Missed Approach

In this exploration task, we started in V-Analytics and selected the trajectories that landed at the end of their observed trips, to filter out noisy and incomplete data and to eliminate transit flights. In Secondo, we received the identifiers of this subset, and issued the query in Section 3 for Missed Approach. After several cycles of adjusting the parameters, the final query that we have reached is:

\[
\begin{align*}
\text{stconstraint}(&"Close", "Up", \text{vec}(\"abba\",\"aba.b\",\"abab\")), \\
\text{stconstraint}(&"Down1", "Up", \text{vec}(\"aabb\",\"aa.bb\")), \\
\text{stconstraint}(&"Up", "Down2", \text{vec}(\"aabb\",\"aa.bb\")); \\
\text{is_within_total_turn_range}.(\text{Pos at periods theRange}(end("Up"), inst(final(.Position)), TRUE, FALSE), 50.0, 270.0) \\
\text{filter}[\text{isdefined}(\text{Close})] \ldots
\end{align*}
\]

Secondo to eliminate transit flights. In served trips, to filter out noisy and incomplete data and selected the trajectories that landed at the end of their observation. The integration of selected trajectories. The same holds for the query in the following section. Figure 2 illustrates one query result. It shows the altitude curve and the 2D trajectory projection of an aircraft that performed a Missed Approach procedure at the Châteauroux-Déols airport. The green color highlights the Down1 and Down2 events, and blue highlights Up.
4.2 Stepwise-Descent Landing

A similar exploration procedure is used to explore the Stepwise-Descent Landing. The pattern is expressed using the \texttt{stpatternexextendstream} operator as follows:

\begin{verbatim}
...stpatternexextendstream[
  Dive1: .SecondAltitudeDerivative < 0.0,
  Lift: .SecondAltitudeDerivative >= 0.0,
  Dive2: .SecondAltitudeDerivative < 0.0 ;
  stconstraint("Dive1", "Lift", vec("aa.bb")),
  stconstraint("Lift", "Dive2", vec("aa.bb"));
  (end("Lift") - start("Lift")) > OneMinute]
filter(isdefined(.Dive1) and
  (AverageDiveAngle(.Alt atperiods .Lift) < 30.0))...
\end{verbatim}

It is expressed as a sequence of increasing (\textit{Dive1}), decreasing (\textit{Lift}), then again increasing rate of descent (\textit{Dive2}). The \textit{Lift} event is required to last for more than one minute, and the diving angle (i.e., the smallest angle between the altitude tangent and the horizontal) is less than 30°. That is, the aircraft is flying almost horizontal.

We attach a map (Figure 3) and a space-time cube (Figure 4) showing median positions of the horizontal (\textit{Lift}) segments of the landing trajectories and 388 trajectories that own these events. The map shows interesting pattern: such landings occur frequently for flights to Paris from SW, W and NW, and sometimes from SE. The space time cube shows that such patterns occur in the morning and late afternoon around Paris, and during the day in Lyon and Nice.

5. CONCLUSIONS

In this paper, we have demonstrated a novel work of integrating a visual analytics system and a moving objects database system. Now we have an integrated system that allows flexible query processing and is open for extensions. It is scalable and can be applied, in principle, to very large data sets. Experiments with larger data sets, and other kinds of operations (e.g., kNN and spatial join) are in preparation.

6. REFERENCES