(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 035 314 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
22.06.2016 Bulletin 2016/25

(51) Int Cl.:
G08G 1/01 *(2006.01)*

(21) Application number: **14199054.9**

(22) Date of filing: **18.12.2014**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(71) Applicant: **Be-Mobile NV**
9090 Melle (BE)

(72) Inventors:
- **Logghe, Steven**
  9090 MELLE (BE)
- **Vanbelle, Wim**
  9090 MELLE (BE)
- **Yperman, Isaak**
  9090 MELLE (BE)
- **Scheerlinck, Karolien**
  9090 MELLE (BE)
- **Houbraken, Maarten**
  9090 MELLE (BE)
- **Vandenberghe, Wim**
  9090 MELLE (BE)
- **Callewaert, Pieter**
  9090 MELLE (BE)
- **De Rijck, Bert**
  9090 MELLE (BE)
- **De Valck, Jens**
  9090 MELLE (BE)
- **Pauwels, Pieter**
  9090 MELLE (BE)
- **Vanlommel, Mario**
  9090 MELLE (BE)

(74) Representative: **Plas, Axel Ivo Michel**
IP HILLS NV
Hubert Frère-Orbanlaan 329
9000 Gent (BE)

(54) A traffic data fusion system and the related method for providing a traffic state for a network of roads

(57) A traffic data fusion system (1) adapted to produce a traffic state (2) for a network of roads, comprising:
- interfaces (100) receiving traffic data (3) from multiple traffic data sources (200);
- a central traffic database (300) collecting said traffic data (3) from said multiple traffic data sources (200);
- a central data fusion engine (400) executing a plurality of traffic state determination algorithms (10) using said traffic data (3) to independently generate respective intermediate traffic states (4); and
- an aggregation engine (500) combining said intermediate traffic states (4) to thereby generate said traffic state (2).

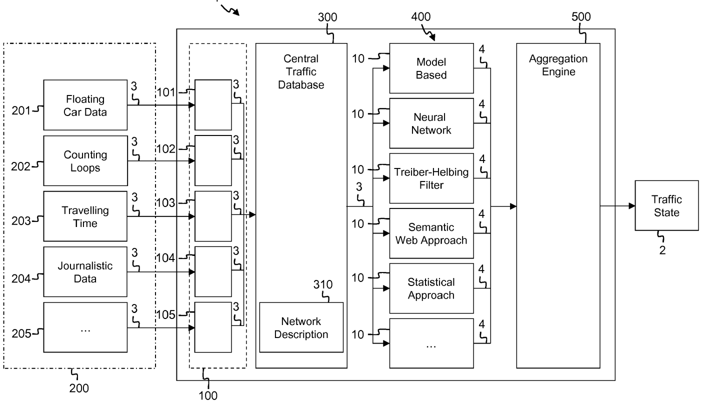


Fig. 1

EP 3 035 314 A1

Description

Field of the Invention

[0001] The present invention relates to the field of processing vehicular traffic data and information for providing an outlook of the traffic state of a network of roads.

Background of the Invention

[0002] Network road capacities in urban and suburban environments insufficiently meet the requirements of current levels and growth rate of traffic volume. Road congestion, phantom jams, reductions in the speed of vehicles due to punctual or fixed obstacles as well as traffic jams resulting from incidents or accidents are examples of traffic situations that need to be well understood in order to provide a wide target audience with traffic alternatives. In order to do so, multiple traffic data sources measure traffic data along main traffic arteries such as highways and roads. The traffic data is collected to generate a traffic state, which gives an overview of the traffic state at the moment the traffic data is collected, and which can be used to predict an overview of the traffic state in the next coming minutes, hours, or days for example. A wide variety of public, such as drivers, truck drivers, engineers and traffic law makers and enforcers depend upon an accurate description of the current and future traffic state situations and scenarios.

[0003] Various prior art techniques for acquiring, analyzing and processing traffic data exist. Prior art techniques typically include calculating velocities of vehicles by for example acquiring series of exact locations of the vehicles along road segments in known intervals. Fixed or static traffic sensors or electronic devices such as video cameras, tag readers, traffic detectors, etc. are for instance installed permanently or temporarily at known locations of main traffic arteries. The sensors relay crossing times of vehicles to a computerized central traffic data and information handling system that calculates the speed of the vehicles between two sensors and generates a traffic state offering for instance an overview of the traffic to an end-user at the moment of the measurement and/or for instance a prediction of the traffic state.

[0004] The patent application US2002/0026278 from Estimotion Inc., entitled "Method and system for modelling and processing vehicular traffic data and information and applying thereof", filed on August 28th 2001 and published on February 28th 2002, describes a method and a system for modelling and processing vehicular traffic data information to provide a single complete current vehicular traffic situation picture. The method comprises acquiring vehicular traffic data from a plurality of sources by tracking a sample of mobile sensors using techniques based on GPS and/or cellular telephone types of wireless communication networks or systems. The acquired vehicular traffic data is then prioritized and filtered in order not to take data emerging from irrelevant sensors and erroneous data into account. The path of a given vehicle is identified along the network of roads and the vehicular traffic data is acquired in known time intervals. The velocities of all mobile sensors that travelled on a specific road segment of the path during a time period of the assessment yields a normalized travel time value, also referred to as NTT value, on that specific road segment. The normalized travel time refers to a travel time normalized with respect to a pre-determined distance, for example, with respect to a distance having a range between about 10 meters to about 100 meters. In other words, the method comprises calculating a mean normalized travel time value, also referred to as mean NTT value, for each road segment of the network of roads using the prioritized and filtered vehicular traffic data and information associated with each source. This results in the generation of a partial current picture of the vehicular traffic situation associated with each source. The system further fuses the partial current traffic situation pictures associated with each source in order to generate a single complete current vehicular traffic situation picture associated with the network of roads. The method also comprises predicting a fixture complete vehicular traffic situation picture associated with the network of roads. The unification of the individual NTT values into a determined value per road segment is done with consideration of a confidence factor of each of the individual data, defined in function of for example the accuracy of the sensor, the amount of footprints, the error rate, etc. Also, the position of the vehicle on a road segment taken into account in the determination of the partial current traffic situation pictures is obtained by introducing assumptions such as the minimal acceleration and the minimal velocity of the vehicle. The accuracy achievable with the described method is therefore limited, as the calculation of the NTT only relies on the fusion of partial current traffic situation pictures associated with sources. Errors generated during the acquisition of the vehicular traffic data as well as errors during the determination of partial current traffic situation pictures are directly transferred to the determination of the single complete current vehicular traffic situation picture and severely impact the quality and the accuracy of the traffic situation picture. This threatens the relevance and the reliability of the traffic state provided to an end-user of the system.

[0005] The patent application US8437948 from Enjoyor Company Limited, entitled "Urban traffic state detection based on support vector machine and multilayer perception", filed on October 18th 2012 and published on April 24th 2014, describes a method and a system employing fusion strategies to compensate for data acquisition and classification errors caused by noise for example. The system relies on the fusion of a support vector machine, also referred to as SVM, and multilayer perceptron, also referred to as MLP, classification algorithms to design a two-tier cascaded classifier. The cascaded two-tier classifier is used to infer a traffic state associated with a road segment during a predefined time interval.

The MLP classification component indeed classifies the data as being in the congested state or non-congested state. The SVM classification component classifies the data as being in the unimpeded state or the busy state for example by utilizing training samples of vehicular traffic data. During the training, weighting of input parameters, for example the average vehicle speed, traffic volume, the average arrival time, can be set and SVM parameters are optimized. The two outputs of the SVM and the MLP classification algorithms are fed to a voting component, which records the detection accuracy of the classifiers and then determines the current traffic state. For example, if the voting component determines that the classifiers have the same detection decision, the voting component outputs the decision as the final state. Alternatively, if the voting component determines that the classifiers each generate different decisions, one decision of a classifier is selected and is treated as the current traffic state. In other words, different classification methods are used to determine a traffic state, but the voting component selects one of them and outputs it as the current traffic state. The relevance and the reliability of the obtained current traffic state are therefore limited, as the determination of the current traffic state only relies on the selection of one traffic state determined by one or more classifiers. The efficiency of the determination of the current traffic state is also low. Indeed, the system generates one or more traffic states, and therefore increases the waste of processing power and the costs associated with the implementation of the method as only one is selected as the output current traffic state.

[0006] It is an objective to disclose a system and the related method that overcome the above identified shortcomings of existing solutions. More particularly, it is an objective to disclose such a system and method for more reliably and more accurately providing an end-user with a traffic state for a network of roads. It is a further objective to disclose such a system and method that improve the robustness, the reliability and the precision of the determination and the prediction of a traffic state for a network of roads without a need for calibration. It is a further objective to disclose such a system and method to determine and predict a traffic state from several data sources in an accurate, efficient, flexible and consistent manner.

Summary of the Invention

[0007] According to a first aspect of the present invention, the above defined objectives are realized by a traffic data fusion system adapted to produce a traffic state for a network of roads, the traffic data fusion system comprising:

- interfaces adapted to receive traffic data from multiple traffic data sources;
- a central traffic database adapted to collect the traffic data from the multiple traffic data sources;
- a central data fusion engine adapted to execute a plurality of traffic state determination algorithms using the traffic data to independently generate respective intermediate traffic states for the network of roads; and
- an aggregation engine adapted to combine the intermediate traffic states to thereby generate the traffic state.

[0008] Multiple data sources acquire traffic data along a network of roads. The multiple data sources can for example be sensors positioned along the roads of the network of roads, and/or sensors positioned in the vehicles, and/or can be journalistic data sources, and/or weather data sources, etc. The traffic data fusion system according to the invention comprises a plurality of traffic state determination algorithms that run on the central data fusion engine. Each traffic state determination algorithm uses a partial set or the full set of available traffic data collected from the multiple traffic data sources to generate a respective intermediate traffic state. In other words, each intermediate traffic state is generated by a respective algorithm independently. The aggregation engine of the traffic data fusion system combines the intermediate traffic states to generate the traffic state. The traffic data collected from the plurality of traffic data sources is therefore computed according to a plurality of traffic state determination algorithms. Each intermediate traffic state is a representation of the traffic data and of the same traffic state of the network of roads as the other intermediate traffic states, but each intermediate traffic state differs from all the other intermediate traffic states as it depicts one or more characteristics of the traffic data differently from the other intermediate traffic states. The fact that the aggregation engine combines all these different representations of the same traffic data to output one single traffic state drastically increases the relevance and the accuracy of the determination of the traffic state. Indeed, as each intermediate traffic state is representative for a different characteristic of the traffic data and the combination of all these different characteristics in one traffic state improves the relevance, the reliability and the robustness of the traffic state outputted to an end-user of the traffic data fusion system. Flexibility in the determination of a traffic state is also provided by the use of a plurality of traffic state determination algorithms. Depending on the situation depicted by the traffic data and identified by the traffic data fusion system, the traffic data fusion system can select to execute two or more traffic state determination algorithms and to combine the respective intermediate traffic states to generate the traffic state. This reduces the processing power required by the traffic data fusion system and decreases the costs associated with its implementation.

[0009] In accordance with the present invention, an intermediate traffic state is a representation of the traffic state at the moment the traffic data fusion system collects the traffic data. In other words, the intermediate traffic state reflects the traffic situation at the moment the traffic data is collected. In accordance with the present invention, traffic state comprises a predefined set of parameters that determines the traffic on a road segment, alternatively on more than one road segment. The predefined set of parameters comprises at least for example the average speed of a vehicle, but also possibly the type of vehicle, a differentiation between driving lanes of the road segment, etc. Alternatively, the intermediate traffic state is a prediction of the traffic state at a time posterior of the moment the traffic data is collected by the traffic data fusion system. The traffic data fusion system makes a prediction of a future traffic state in time. A plurality of traffic state determination algorithms extrapolate the traffic data to be able to predict the traffic state in the future and to generate an intermediate traffic state representative for a prediction of the traffic state. Alternatively, the intermediate traffic state is a representation and/or an extrapolation of the traffic state at a time anterior of the moment the traffic data is collected by the traffic data fusion system. The traffic data fusion system depicts of a past traffic state in time. A plurality of traffic state determination algorithms extrapolate the traffic data to be able to depict the traffic state in the past and to generate an intermediate traffic state representative for a past traffic state. Interfaces receive traffic data from multiple data sources. To each data source corresponds an interface. Alternatively, an interface is adapted to receive traffic data from a plurality of data sources. An interface is for example a sensor, a sensor adapted for wireless data communication, a phone, a Global Navigation System, a camera, etc.

[0010] According to an optional embodiment, the plurality of traffic state determination algorithms comprise two or more of:

- a Kalman filter, preferably the ensemble Kalman filter or EnKF;
- a Treiber-Helbing filter, preferably the Extended Generalized Treiber-Helbing filter or EGTF;
- a neural network, preferably a Multi-Layer Feed-forward neural network;
- machine learning;
- pattern recognition;
- a statistic algorithm;
- a genetic algorithm;
- a Lagrange and Euler Cinematic wave method;
- a cumulative number of vehicles;
- a multicriteria method;
- an adaptive smoothing;
- an adaptive Kalman method;
- a particle swarm optimization;
- ant colony optimization;
- a dynamic programming; and
- simulated annealing.

[0011] This way, the traffic state results from the combination of two or more intermediate traffic states independently generated by two or more respective traffic state determination algorithms. Depending on the situation depicted by the traffic data and identified by the traffic data fusion system, the traffic data fusion system can select to execute two or more traffic state determination algorithms and to combine the respective intermediate traffic states to generate the traffic state. The fact that the aggregation engine combines all these different representations of the same traffic data to output one single traffic state drastically increases the relevance, the accuracy and the reliability of the determination of the traffic state. The intermediate traffic states are independently generated by the central data fusion engine. This way, the central data fusion engine selects traffic state determination algorithms to be executed and does not need to perform all the traffic state determination algorithms at all times. This saves processing power and therefore reduces the costs associated with the traffic data fusion system.

[0012] In accordance with the present invention, Kalman filtering, also known as linear quadratic estimation or LQE, is an algorithm that uses a series of measurements observed over time, containing noise and other inaccuracies, and produces estimates of unknown variables. A Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement, corrupted with some amount of error, is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. Because of the algorithm's recursive nature, it can run in real time using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required. Using this filtering technique enables the determination of variables that are not directly observed, such as travel times for example, and allows to deal with all types of traffic situations represented by the traffic data as long as these traffic situations can be described by the model. The Kalman filtering technique is also relevant even if traffic data for a certain road segment or a certain time interval is missing. Indeed, the model is able to fill the gaps by providing a result at any moment and at any place. The Kalman filtering technique further offers the possibility to clarify possible inconsistencies and errors in the traffic data and inconsistencies between traffic data measured on different road segments. Indeed, the fact that the model cannot be applied to the collected traffic data can be an indication that the traffic data is not consistent, or that the traffic data is not consistent with traffic engineering laws. It is also possible to extrapolate the model in order to predict a future traffic state when the future traffic state is brought close to reality using traffic data. Another advantage of the Kalman filtering method is that it provides insight into the road segment from which traffic data should be measured in order to optimize the determination of the traffic state, i.e. rely on an optimal overview of the traffic data. According to the present invention, the state of the traffic at the time t is called $x(t)$. The Kalman filtering technique is able to predict the state of the system at any point in time $t+\Delta t$ according to the function: $x(t+\Delta t)=M(x(t), u(t), p)$, where $x(t)$ is the current traffic state, where $u(t)$ is the external control of the system, for example the boundary conditions of the model, and where p is a time-independent parameter. The data assimilation techniques according to the present invention are sequential data assimilation techniques. Starting from a state $x(t)$, the model is used to estimate the traffic state at the time $t+\Delta t$, for which the measurements $y(t+\Delta t)$ are available. The prediction of the model $x^f(t+\Delta t)$ is then combined with the $y(t+\Delta t)$ in an improved prediction $x^a(t+\Delta t)$ in an assimilation step. Alternatively, it is also possible to fit the external control $u(t+\Delta t)$ during the assimilation step.

[0013] In accordance with the present invention, the alternative ensemble Kalman filter, or EnKF, is used to continue to assimilate the measurements. EnKF is a popular alternative to the Kalman filtering method, and is used with large non-linear models. The EnKF algorithm uses a plurality of model realizations to depict the uncertainty of the model. The new state $x^a(t+\Delta t)$ is then determined by adjusting the model prediction (including the uncertainty) according to the measurements (including the measurement uncertainty). If the model shows less uncertainty than the measurement, then the adjusted prediction resembles the measurements. During the adjustment of the prediction, a new estimation of the remaining model uncertainty is performed.

[0014] In accordance with the present invention, another method for data fusion is the use of a Treiber-Helbing filter. This filter exploits the fact that, in congested traffic, perturbations travel upstream at a near-constant speed, while in free traffic information propagates downstream. As in the first order kinematic wave traffic flow model, it is assumed that traffic data propagates along with a speed which is equal to the slope in the basic diagram of the traffic flow theory. As a result, one obtains velocity, flow, or other variables as smooth functions of space and time. The parameters that characterize a traffic state, for example the speed and/or the density of vehicles, are interpolated with this filter, so that measurements that are acquired recently and that are acquired for a close road segment are associated with a heavier weight than other measurements. Data sources that comprise more data traffic can also be more heavily weighted in the determination of the traffic state than other data sources which comprise less traffic data. To be able to use more than one data source, this filter is extended to the Extended Generalized Treiber-Helbing filter, also referred to as EGTF. The EGTF is able to fuse multiple data sources, as long as for each of these, it is possible to estimate under which traffic regime, for example free flowing or congested, the traffic data was collected. The main advantage of this approach is that it allows merging or fusing traffic data from different data sources, characterized by different spatial and temporal resolutions and/or different accuracies and/or different reliabilities, into a consistent, coherent and meaningful traffic information. This process allows the use of the theoretical traffic laws.

[0015] In accordance with the present invention, another method for data fusion is machine learning. Machine learning is a subfield of computer science and statistics that deals with the construction and study of systems that can learn from data, rather than follow only explicitly programmed instructions. Machine learning tasks can be of several forms. In supervised learning, the computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. In unsupervised learning, no labels are given to the learning algorithm, leaving it on its own to form groups of similar inputs, density estimates or projections of high-dimensional data that can be visualised effectively. In reinforcement learning, a computer program interacts with a dynamic environment in which it must perform a certain goal, for example driving a vehicle, without a teacher explicitly telling it whether it has come close to its goal or not. Specific algorithms that follow this methodology are for example neural networks and pattern recognition, which will be discussed in more detail below.

[0016] In accordance with the present invention another method for data fusion is the use of neural networks. Neural networks belong to the category of machine learning algorithms. Neural networks are very suited to perform specific tasks such as pattern recognition, perception and control. Neural networks compute these tasks very fast and accurately. Neural networks, also referred to as Biological Neural Networks or BNNs have the further advantage that they are robust, i.e. that data provided with a lot of noise is - to some extent - processed in a correct manner. By imitating the brain processes and the brain architecture, specific properties are used by the neural networks to execute brain-related tasks. These simplified BNNs are called Artificial Neural Networks, also referred to as ANNs. BNNS and ANNs both comprise parallel units called neurons. A Multi-Layer Feed-forward neural network is an artificial neural network where connections between the units do not form a directed cycle. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes, if any, and to the output nodes. There are no cycles or loops in the network. The imitation lies in the fact that signals that were first non-linearly processed are passed on to all the forward-associated neurons. An ANN for example complies with the following steps: signals enter a network via an input layer comprising, for example, n neurons. The signals are processed non-linearly and passed on to all the forward associated units in the following layer: this can for example be a hidden layer, but also the output layer. These operations are repeated until the output layer having m neurons has been reached. Then the ANN spreads the signals again. The description shows that the adaptation of an ANN is actually nothing more than a non-linear mapping of an *n*-dimensional input vector, often referred to as pattern, to an m-dimensional output vector. The way the mapping is achieved is determined by the choice of network architecture, the choice of the transfer function within the artificial neurons and the selection of the learning rule which is in force during the learning process. The network architecture depends on the number of artificial neurons per layer and the connections of neurons between layers and the connection of neurons in the same layers. The selection of the transfer function is always an increasing function. The choice of the transfer function may be per neuron or per layer, but is usually the same for each neuron of the entire network. The selection of the learning rule determines how the synaptic weights, with which incoming signals are multiplied, change during the learning process. The use of a neural network is justified by its short development time, its robustness with respect to the collection of noisy data, its efficiency with respect to the computer processing time and its adaptability in a changing environment. In the case of a Multi-Layer Feedforward neural network, the ANNs are trained under supervision. This training comprises three phases: the learning phase, also called the training phase, during which training patterns are introduced along with end purposes. The difference between the end purpose and the output generated by the ANN is an error term that is a measure used to adjust the synaptic weights and therefore to minimize the cumulative error term of all the training patterns. During the second phase, called the test phase, training patterns are provided to the ANN. Judging from the cumulative error term, one can estimate how generalizing the ANN is. In the case that it is small enough, the last phase, called the validation phase, can be reached where the ANN is confronted with the absence of end purposes to which its calculation can be compared to. The ANN must then indeed generate them itself. In accordance with the present invention, the traffic data is therefore divided into four sets of traffic data. Two sets of traffic data are used to train the ANN, one set is used during the test phase and the last set is used during the validation phase. Relying on cross-correlation, it is then possible to determine the result for the whole traffic data.

[0017] In accordance with the present invention, another method for data fusion is pattern recognition. Pattern recognition also belongs to the category of machine learning algorithms. This branch of artificial intelligence focuses on the recognition of patterns and regularities in data. In many cases, these patterns are learned from labelled "training" data, but when no labelled data are available other algorithms can be used to discover previously unknown patterns. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of classes. However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labelling, which assigns a class to each member of a sequence of values; and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence. Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs, taking into account their statistical variation. Pattern recognition can thus be used to combine multiple data sources into a single accurate description of the traffic state.

[0018] In accordance with the present invention, another method for data fusion is the use of a genetic algorithm. A genetic algorithm is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization problems and search problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. This algorithm can be used to minimize the difference between the fused traffic state and reality.

[0019] In accordance with the present invention, another method for data fusion is the use of the Lagrange and Euler Cinematic wave method. The kinematic wave model is often used in simulation tools to describe dynamic traffic flow and to estimate and predict traffic states. Discretization of the model is generally based on Eulerian coordinates, which are fixed in space. However, the Lagrangian coordinate system, in which the coordinates move with the velocity of the vehicles, results in more accurate solutions. Furthermore, if the model includes multiple user classes, it describes real traffic more accurately. Such a multiclass model, in contrast to a mixed-class model, treats different types of vehicles, for example passenger cars and trucks or vehicles with different origins or destinations, or both, differently. The Lagrangian coordinate system is combined with a multiclass model, and a Lagrangian formulation of the kinematic wave model for multiple user classes is proposed. It is shown that the advantages of the Lagrangian formulation also apply for the multiclass model. Simulations based on the Lagrangian formulation result in more accurate solutions than simulations based on the Eulerian formulation.

[0020] In accordance with the present invention, another method for data fusion is the use of cumulative number of vehicles such as the Link Transmission Model. The Link Transmission Model, also referred to as LTM, is a Dynamic Network Loading model for a macroscopic simulation-based Dynamic Traffic Assignment model; vehicles are moved as a continuum. In LTM, traffic propagation on network links is consistent with kinematic wave theory. This theory provides substantial realism in the representation of queue-propagation and queue-dissipation. Furthermore, LTM considers a detailed description of traffic dynamics at signalized and un-signalized intersections. Local flow restrictions and experienced intersection delays are consistent with state-of-the-art queuing theory. Since the LTM solution algorithm is computationally efficient and walks through simulations in large time steps, large scale networks can be dealt with in a small amount of time.

[0021] In accordance with the present invention, another method for data fusion is the combination of the Extended Generalized Treiber-Helbing Filter, also referred to as EGTF, and the traffic conservation law. The EGTF is able to fuse multiple data sources, in order to estimate the traffic state (velocity, density and intensity) as a function of space and time. This filter encompasses the fundamental concepts of traffic flow theory, namely the upstream propagation of congested traffic perturbations and the downstream propagation of free flow traffic waves. The conservation law, on the other hand, states that in case of a closed section, no on- or off-ramps are present, the number of vehicles cannot change in time and space. In order to apply this law, the cumulative number of vehicles, for example determined with the Link Transmission Model, also referred to as LTM, is used. During the process of traffic estimation with the EGTF, the conservation law is applied to simultaneously correct the estimated traffic state, which can be necessary in case the original data sources contains errors. This extension of the EGTF thus allows us to minimize the effect of false measurements, e.g. due to technical failure, in the original available data sources, which empowers the original EGTF considerably.

[0022] In accordance with the present invention, another method for data fusion is the use of an adaptive Kalman method. Kalman filters have been widely used for navigation and system integration. A possible challenge associated with Kalman filters is how to assign suitable statistical properties to both the dynamic and the observational models. For Global Navigation System navigation, the manoeuvre of the vehicle and the level of measurement noise are environmental dependent, and difficult to be predicted. Therefore, to assign constant noise levels for such applications is not realistic. It is demonstrated that adaptive algorithms are more robust to sudden changes of vehicle motion and measurement errors than a conventional Kalman filter is for vehicle navigation. Alternatively, other variants of the implementation of the Kalman filtering method are also in accordance with the present invention.

[0023] In accordance with the present invention, another method for data fusion is the use of a particle swarm optimization. This computational method optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. The method optimizes a problem by having a population of candidate solutions, referred to as particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle’s position and velocity. Each particle’s movement is influenced by its local best known position but is also guided towards the best known positions in the search-space, which are updated as better positions are found by other particles.

[0024] In accordance with the present invention, another method for data fusion is the use of an ant colony optimization algorithm. This algorithm is a probabilistic technique for solving computational problems which can be reduced to searching for an optimal path in a graph.

[0025] In accordance with the present invention, another method for data fusion is the use of a dynamic programming method. This method solves complex problems by breaking them down into simpler subproblems and by then combining the solutions of the subproblems to reach the best overall solution for the problem. The dynamic programming approach seeks to solve each subproblem only once, thus reducing the number of computations. Once the solution to a given subproblem has been computed, it is stored or "memorized": the next time the same solution is needed, it is simply looked up.

[0026] In accordance with the present invention, another method for data fusion is the use of a simulated annealing method. This method locates a good approximation to the global optimum of a given function in a large search space. The algorithm relies on a slow decrease in the probability of accepting worse solutions as it explores the solution space.

[0027] According to an optional embodiment, the aggregation engine is adapted to calculate a mean value of the intermediate traffic states as the traffic state.

[0028] This way, before the aggregation engine is applied, the outputs of the different data fusion algorithms are validated through comparison. Extreme outliers can then be deleted before the aggregation engine combines the results into the traffic state.

[0029] This way, the traffic state takes all the available intermediate traffic states independently resulting from their respective traffic state determination algorithms into account. All the intermediate traffic states are associated with a same weight value during the determination of the traffic state, and the traffic state is obtained by averaging all the available intermediate traffic states. All the available intermediate traffic states are combined and the resulting traffic state is divided by the total number of available intermediate traffic states, thereby generating the traffic state. In other words, each intermediate traffic state is considered with the same importance during the determination of the traffic state and each intermediate traffic state has the same impact as all the other intermediate traffic states on the determination of the traffic state. This combination of several intermediate traffic states drastically increases the accuracy and the reliability of the traffic state. Indeed, each intermediate traffic state is obtained from a different traffic state determination algorithm and therefore is a different representation of the traffic data compared to all the other intermediate traffic states. The determination of the traffic state therefore relies on a robust and relevant combination of all these different representations of the same traffic data. Alternatively, in the case that the traffic data fusion system performs a prediction in time of the traffic state, the fact that the aggregation engine calculates a mean value of the intermediate traffic states as the traffic state drastically increases the reliability and the accuracy of the prediction. The probability that the generated predicted traffic state becomes real increases.

[0030] According to an optional embodiment, the aggregation engine is adapted to calculate a median value of the intermediate traffic states as the traffic state.

[0031] This way, the traffic state is determined by identifying the median value of all the available intermediate traffic states. A calculation is performed to identify the median value of all the different representations of the traffic data represented by the respective different intermediate traffic states. This makes the determination of the traffic state more relevant as the system is able to identify similarities between all the different independently generated intermediate traffic states and to associate these similarities with a larger weight during the generation of the traffic state. This way, similar representations of traffic data identified on different intermediate traffic states are considered as more relevant for the determination of the traffic state than representations and characteristics of the traffic data that are only present in a minority of intermediate traffic states. This improves the accuracy of the traffic state as the traffic state is calculated to be as close as possible to the current traffic state and/or to the most likely predicted traffic state.

[0032] According to an optional embodiment, the aggregation engine is adapted to calculate a weighted sum of the intermediate traffic states as the traffic state.

[0033] This way, all the intermediate traffic states that are generated by respective traffic state determination algorithms are taken into account for the generation of the traffic state. Some intermediate traffic states are associated with a higher weight value that others, which increases the influence and the impact of the intermediate traffic states associated with a higher weight value on the generation of the traffic state. This increases the reliability of the traffic state. The fact that all the intermediate traffic states are taken into account during the generation of the traffic state drastically increases the accuracy and the robustness of the generation of the traffic state.

[0034] According to an optional embodiment, the weighted sum comprises weights representative for the reliability of the respective traffic state determination algorithms.

[0035] This way, all the intermediate traffic states that are independently generated by respective traffic state determination algorithms are taken into account for the generation of the traffic state. This drastically increases the accuracy and the robustness of the generation of the traffic state. Furthermore, each intermediate traffic state is associated with a weight representative for the reliability of the respective traffic state determination algorithms. The traffic data fusion system is able to evaluate the reliability of two or more traffic state determination algorithms for the traffic state situation depicted by the traffic data it collected. In other words, depending on the nature of the traffic data and of the situation the traffic data reflects, the traffic data fusion system determines the reliability of two or more of the traffic determination algorithms by for example associating a reliability factor to two or more of the traffic state determination algorithms. As soon as the traffic data received and collected by the traffic data fusion system is updated, the traffic data fusion system re-evaluates the reliability of two or more of the traffic state determination algorithms for the situation reflected by the traffic data. Reliable traffic state determination algorithms are then associated with weights demonstrating a higher numerical value than the weights associated with traffic state determination algorithms identified as less reliable. This way, intermediate traffic states generated by reliable traffic state determination algorithms are weighted with a higher numerical value than other intermediate traffic states during their combination, i.e. during the generation of the traffic state. In other words, the more reliable a traffic state determination algorithm is, the higher the numerical value of the weight associated with the corresponding intermediate traffic state is, and the larger the impact of the intermediate traffic state on the generation of the traffic state. The traffic state is therefore more influenced by reliable intermediate traffic states, which drastically increases the reliability and the relevance of the traffic state. Alternatively, in the case that the data fusion system performs a prediction of the traffic state in time, the fact that the reliability of the traffic state determination algorithm is taken into account during the generation of the traffic state drastically increases the reliability and the accuracy of the prediction. The resulting probability that the generated predicted traffic state is realized therefore increases.

[0036] According to an optional embodiment, the traffic state comprises one or more of:

- a vehicle volume;
- a vehicle classification;
- a vehicle density;
- an average vehicle speed; and
- journalistic traffic data.

[0037] This way, the traffic state can be used to reflect the state of the network of roads, for example if the network of roads is free-flowing or if the network of roads is congested. This traffic state makes information over vehicles on the network of roads available and the traffic state also reflects journalistic traffic data acquired. This information can for example then be shared with other vehicles and/or data sources. Further traffic parameters can be deduced from the traffic state, for example the travel time, the gap distance between two vehicles, the vehicle heading, the delay, the level of service, the presence of absence of a traffic jam, etc. Indicators such as the level of gas emission of a vehicle, the level of noise generated by a vehicle, hours during which a vehicle consumes the most, etc can also be deduced from the traffic state. The traffic state facilitates the representation and the visualization of the traffic data by an end-user of the traffic data fusion system, and it therefore facilitates a diagnosis performed by the end-user to identify if the network of roads is free-flowing or if it is congested. It also reduces the time needed for an end-user to analyze the traffic state and to identify the state of the network of roads. The average vehicle speed and/or the journalistic traffic data can for example be used, possibly in combination with other parameters, when predicting the traffic state in time.

[0038] According to an optional embodiment, the traffic state is validated. This validation is performed simultaneously with the traffic estimation, this means directly after publishing the traffic state and before publishing the next traffic state. The validation result is used to adapt the weights given to the different data fusion algorithms. The validation result can for example indicate that in congestion, the results obtained with the Treiber-Helbing Filter are significantly more accurate than with neural networks. Thus, based on this validation, the weight of the Treiber-Helbing Filter can be increased with respect to the weight of neural networks. In addition, the validation result can also be used to determine the reliability of the different data sources. With these results, it can be decided to give different weights to the different data sources. In this way, the complete data fusion methodology can be trained.

[0039] According to an optional embodiment, the traffic state is periodically updated.

[0040] Traffic data is periodically updated. This way, the traffic state is periodically updated, which means that the traffic state remains relevant in time as the traffic state depicts and reflects recently acquired traffic data. The fact that the traffic state is periodically updated also ensures the robustness of the traffic state. An end-user of the traffic data fusion system is indeed provided with a real-time viewing experience of the state of the network of roads. The periodic update also increases the relevance, the reliability and the accuracy of the prediction of the traffic state in time as the determination of the traffic state can more rapidly take changes of the traffic data into account and therefore adapt the prediction of the traffic state in time accordingly. For example, the traffic state can be updated every 15 seconds, or every 30 seconds, or every 45 seconds, or every minute, or every day, etc.

[0041] According to an optional embodiment, the traffic state is calculated for road segments of a predetermined length.

[0042] Traffic data is collected for each individual lane of every road segment of the network of roads. The predetermined length of a road segment can for example be substantially equal to 25 meters, or 50 meters, or 100 meters, or 1 kilometer, etc. Every road segment of the network of roads has the same predetermined length. Alternatively, at least two road segments of the network of roads have different predetermined lengths. This way, the traffic data is regularly updated and the traffic state therefore reflects a recent and reliable state of the road segment. This increases the reliability and the robustness of the traffic state.

[0043] According to an optional embodiment, the multiple traffic data sources comprise two or more of:

- a Floating Car Data system or FCD system;
- a point measurement system;
- a trajectory measurement system;
- a journalistic data system;
- a weather data system; and
- a traffic management data system.

[0044] This way, the traffic state is generated relying on several data sources. The plurality of inputs to the traffic data fusion system increases the accuracy and the reliability of the determination of the traffic state. In accordance with the present invention, Floating Car Data, also known as FCD or floating cellular data, is a method to determine the traffic speed on the road network. It is based on the collection of for example localization data, speed, direction of travel and time information from mobile phones in vehicles that are being driven. These traffic data are the essential source for traffic information and for most intelligent transportation systems. This means that every vehicle with for example an active mobile phone acts as a sensor for the road network. Based on these data, traffic congestion can be identified, travel times can be calculated, and traffic reports can be rapidly generated. This way, no additional hardware on the network of roads is necessary, which drastically reduces the costs associated with the traffic data collection. In accordance with the present invention, a point measurement system performs a measurement locally, either in a vehicle or on one road segment of the network of roads. In accordance with the present invention, a trajectory measurement is performed by measuring the position of vehicles at different positions, for example on several road segments of the network of roads and determining the corresponding trajectories. In accordance with the present invention, journalistic data is provided to the traffic fusion data system. Journalistic data gathers input from for example users of vehicles, such as internet messages, tweets, location updates, SMS, social media, and/or even traffic directly provided by the vehicles themselves. Weather data system identifies the weather conditions on the road segments of the network of roads. The fact that the weather conditions, for example the presence of rain, fog, snow, the temperature, etc. are taken into account in the generation of the traffic state increases the reliability of the traffic state. Traffic management systems comprise for example traffic cameras, number plate recognition systems, and induction loops embedded in the road segments of the network of roads. The combination and cross-comparison of traffic data acquired from vehicles and of traffic data acquired on the road segments themselves increases the accuracy of the determination of the traffic state.

[0045] According to a second aspect of the present invention, there is provided a method for producing a traffic state for a network of roads, the method comprising the steps of:

- receiving traffic data from multiple traffic data sources;
- collecting the traffic data from the multiple traffic data sources;
- executing a plurality of traffic state determination algorithms using the traffic data to independently generate respective intermediate traffic states for the network of roads; and
- combining the intermediate traffic states to thereby generate a traffic state.

[0046] Multiple data sources acquire traffic data along a network of roads. The multiple data sources can for example be sensors positioned along the roads of the network of roads, and/or sensors positioned in the vehicles, and/or can be journalistic data sources, and/or weather data sources, etc. Each traffic state determination algorithm uses a partial set or the full set of available traffic data collected from the multiple traffic data sources to independently generate a respective intermediate traffic state. In other words, each intermediate traffic state is independently generated by a respective algorithm. The intermediate traffic states are then combined to generate the traffic state. The traffic data collected from the plurality of traffic data sources is therefore computed according to a plurality of traffic state determination algorithms. Each intermediate traffic state is a representation of the traffic data and of the same current traffic state of the network of roads than the others intermediate traffic state, but each intermediate traffic state differs from all the other intermediate traffic states as it depicts one or more characteristics of the traffic data differently than the other intermediate traffic states. The fact that the all these different representations of the same traffic data are combined to output one single traffic state drastically increases the relevance and the accuracy of the determination of the traffic state. Indeed, as each intermediate traffic state is representative for a different characteristic of the traffic data and the combination of all these different characteristics in one traffic state improves the relevance, the reliability and the robustness of the traffic state outputted to an end-user of the traffic data fusion system. Flexibility in the determination of a traffic state is also provided by the use of a plurality of traffic state determination algorithms. Depending on the situation depicted by the traffic, two or more traffic state determination algorithms can be selected and executed independently and the generated respective intermediate traffic states are combined to generate the traffic state. This reduces the processing power required to reach the traffic state and decreases the costs associated with the implementation of the method.

[0047] In accordance with the present invention, an intermediate traffic state is a representation of the traffic state at the moment the traffic data fusion system collects the traffic data. In other words, the intermediate traffic state reflects the traffic situation at the moment the traffic data is collected. Alternatively, the intermediate traffic state is a prediction of the traffic state at a time posterior of the moment the traffic data is collected by the traffic data fusion system. The traffic data fusion system makes a prediction of a future traffic state in time. A plurality of traffic state determination algorithms extrapolate the traffic data to be able to predict the traffic state in the future and to generate an intermediate traffic state representative for a prediction of the traffic state. Alternatively, the intermediate traffic state is a representation and/or an extrapolation of the traffic state at a time anterior of the moment the traffic data is collected by the traffic data fusion system. The traffic data fusion system depicts of a past traffic state in time. A plurality of traffic state determination algorithms extrapolate the traffic data to be able to depict the traffic state in the past and to generate an intermediate traffic state representative for a past traffic state.

[0048] The current invention in addition also relates to a computer program comprising software code adapted to perform the method according to the present invention.

[0049] The invention further relates to a computer readable storage medium comprising the computer program according to the present invention.

Brief Description of the Drawings

[0050]

Fig. 1 schematically illustrates an embodiment of a traffic data fusion system adapted to produce a traffic state.

Fig. 2 schematically illustrates an embodiment of a traffic data fusion system adapted to produce a traffic state and comprising a validation unit A adapted to compare the traffic state to the traffic data.

Fig. 3 schematically illustrates an embodiment of a traffic data fusion system adapted to produce a traffic state and comprising a validation unit B adapted to compare the traffic state determination algorithms.

Fig. 4 schematically illustrates an embodiment of a traffic data fusion system adapted to produce a traffic state and comprising a validation unit A adapted to compare the traffic state to the traffic data and a validation unit B adapted to compare the traffic state determination algorithms.

Fig. 5 schematically illustrates an embodiment of the generation of the traffic state from a plurality of weighted intermediate traffic states.

Fig. 6 schematically illustrates a suitable computing system for hosting the system of Figure 1.

Detailed Description of Embodiment(s)

[0051] According to an embodiment shown in Fig. 1, a traffic data fusion system 1 comprises a plurality of interfaces 100, a plurality of data sources 200, a central traffic database 300, a central data fusion engine 400, and an aggregation engine 500. Traffic data 3 is collected by a plurality of data sources 200. For example in Fig. 1, Floating Car Data source 201, counting loops source 202, travelling time source 203, journalistic data source 204 and/or any other suitable data source 205 receive and collect traffic data 3 for a network of roads. Interfaces 100 receive the traffic data 3 from the plurality of data sources 200. As visible in Fig. 1, each interface 100 receives traffic data 3 from one data source 200. In Fig. 1, interface 101 to interface 105 respectively receive traffic data 3 from data source 201 to data source 205. According to an alternative embodiment, each interface 100 receives traffic data 3 from more than one data source 200. The central traffic database 300 collects the traffic data 3 from the interfaces 100. The central traffic database 300 comprises a road network description 310. A road network description 310 consists of a consistent logical description of the location, properties and topological model of the road network. It contains a set of nodes and links with additional properties described in segments. A node describes a geographical point with coordinates, for example a longitude and a latitude, in a world surface coordinate system. It contains a node ID and additional node properties. Roads are described as links, a vector or connection between different nodes. Links have driving properties like driving direction, number of lanes, speed limits, road length, etc. A set of links and nodes describes a graph that can be used in a route planner or a Geographic Information System. A link can be split in different short sections, referred to as segments. According to an alternative embodiment, the central traffic database 300 comprises the interfaces 100. The central data fusion engine 400 uses the traffic data 3 to generate intermediate traffic states 4. The central data fusion engine 400 performs a plurality of traffic state determination algorithms 10 using the traffic data 3 to independently generate respective intermediate traffic states 4 for the network of roads. The traffic state determination algorithms 10 visible in Fig. 1 are for example model based, neural network, Treiber-Helbing filter, a semantic web approach, a statistical approach, etc. The aggregation engine 500 receives the intermediate traffic states 4 generated by the central data fusion engine 400 and combines the intermediate traffic states 4 to generate the traffic state 2.

[0052] According to an embodiment shown in Fig. 2, a traffic data fusion system 1 comprises a plurality of interfaces 100, a plurality of data sources 200, a central traffic database 300, a central data fusion engine 400, an aggregation engine 500 and a validation unit 7 labelled A. Traffic data 3 is collected by a plurality of data sources 200. For example in Fig. 2, Floating Car Data source 201, counting loops source 202, travelling time source 203, journalistic data source 204 and/or any other suitable data source 205 receive and collect traffic data 3 for a network of roads. Interfaces 100 receive the traffic data 3 from the plurality of data sources 200. As visible in Fig. 2, each interface 100 receives traffic data 3 from one data source 200. In Fig. 2, interface 101 to interface 105 respectively receive traffic data 3 from data source 201 to data source 205. According to an alternative embodiment, each interface 100 receives traffic data 3 from more than one data source 200. The central traffic database 300 collects the traffic data 3 from the interfaces 100. The central traffic database 300 comprises a road network description 310. A road network description 310 consists of a consistent logical description of the location, properties and topological model of the road network. It contains a set of nodes and links with additional properties described in segments. A node describes a geographical point with coordinates, for example a longitude and a latitude, in a world surface coordinate system. It contains a node ID and additional node properties. Roads are described as links, a vector or connection between different nodes. Links have driving properties like driving direction, number of lanes, speed limits, road length, etc. A set of links and nodes describes a graph that can be used in a route planner or a Geographic Information System. A link can be split in different short sections, referred to as segments. According to an alternative embodiment, the central traffic database 300 comprises the interfaces 100. The central data fusion engine 400 uses the traffic data 3 to generate intermediate traffic states 4. The central data fusion engine 400 performs a plurality of traffic state determination algorithms 10 using the traffic data 3 to independently generate respective intermediate traffic states 4 for the network of roads. The traffic state determination algorithms 10 visible in Fig. 2 are for example model based, neural network, Treiber-Helbing filter, a semantic web approach, a statistical approach, etc. The aggregation engine 500 receives the intermediate traffic states 4 generated by the central data fusion engine 400 and combines the intermediate traffic states 4 to generate the traffic state 2. As visible in Fig. 2, the system 1 further comprises a validation unit 7 labelled A. The validation unit 7 labelled A is adapted to compare the published traffic state 2 with the reality of the traffic state depicted by the traffic data 3 collected by the data sources 200. The validation unit 7 labelled A therefore receives the traffic state 2 and the data 3 as inputs. The validation unit 7 labelled A outputs a comparison 8 of how reliable the published traffic state 2 is compared to the traffic data 3 collected by the data sources 200. The aggregation engine 500 receives the comparison 8 generated by the validation unit 7 labelled A. The aggregation engine 500 takes this comparison 8 into account when determining the weights of the different traffic state determination algorithms 10 in order to alter the reliability of the data sources 200. This way, predictions of the traffic state 2 generated by the traffic data fusion system 1 can be validated by the validation unit 7 labelled A. The weights of one or more traffic state determination algorithms 10 can be adapted so that the traffic state 2 is more in accordance with the traffic data 3. Alternatively, the comparison 8 generated by the validation unit 7 labelled A is used to allocate weights to the traffic data 3. These weights are a measure of the reliability of each of the data sources 200. For example, if the validation unit 7 labelled A identifies that the traffic state 2 does not have any correspondence with the traffic data 3 collected by a data source 200, the validation unit 7 labelled A will generate a weight for the data source 200 smaller than the weights associated with other data sources 200. This way, the validation unit 7 labelled A ensures the traffic data 2 is more in accordance with the traffic data 3 by allocating more importance to one or more data sources 200, i.e. by allocating weights demonstrating a higher numerical value to data sources 200 from which the traffic data 3 is in accordance with the traffic state 2 and/or by respectively allocating weights demonstrating a lower numerical value to data sources 200 from which the traffic data 3 is not in accordance with the traffic state 2.

[0053] According to an embodiment shown in Fig. 3, a traffic data fusion system 1 comprises a plurality of interfaces 100, a plurality of data sources 200, a central traffic database 300, a central data fusion engine 400, an aggregation engine 500 and a validation unit 9 labelled B. Traffic data 3 is collected by a plurality of data sources 200. For example in Fig. 3, Floating Car Data source 201, counting loops source 202, travelling time source 203, journalistic data source 204 and/or any other suitable data source 205 receive and collect traffic data 3 for a network of roads. Interfaces 100 receive the traffic data 3 from the plurality of data sources 200. As visible in Fig. 3, each interface 100 receives traffic data 3 from one data source 200. In Fig. 3, interface 101 to interface 105 respectively receive traffic data 3 from data source 201 to data source 205. According to an alternative embodiment, each interface 100 receives traffic data 3 from more than one data source 200. The central traffic database 300 collects the traffic data 3 from the interfaces 100. The central traffic database 300 comprises a road network description 310. A road network description 310 consists of a consistent logical description of the location, properties and topological model of the road network. It contains a set of nodes and links with additional properties described in segments. A node describes a geographical point with coordinates, for example a longitude and a latitude, in a world surface coordinate system. It contains a node ID and additional node properties. Roads are described as links, a vector or connection between different nodes. Links have driving properties like driving direction, number of lanes, speed limits, road length, etc. A set of links and nodes describes a graph that can be used in a route planner or a Geographic Information System. A link can be split in different short sections, referred to as segments. According to an alternative embodiment, the central traffic database 300 comprises the interfaces 100. The central data fusion engine 400 uses the traffic data 3 to generate intermediate traffic states 4. The central data fusion engine 400 performs a plurality of traffic state determination algorithms 10 using the traffic data 3 to independently generate respective intermediate traffic states 4 for the network of roads. The traffic state determination algorithms 10 visible in Fig. 3 are for example model based, neural network, Treiber-Helbing filter, a semantic web approach, a statistical approach, etc. The aggregation engine 500 receives the intermediate traffic states 4 generated by the central data fusion engine 400 and combines the intermediate traffic states 4 to generate the traffic state 2. As visible in Fig. 3, the system 1 further comprises a validation unit 9 labelled B. The validation unit 9 labelled B is adapted to compare the intermediate traffic states 4 generated by the central data fusion engine 400 and to delete intermediate traffic states 4 the validation unit 9 labelled B identifies as not reliable. The validation unit 9 labelled B therefore receives the intermediate traffic states 4 as input. The validation unit 9 labelled B outputs reliable intermediate traffic states 4. For example, if one intermediate traffic state 4 represents an intermediate traffic state that is completely different, for example a complete opposite representation and/or prediction than the one of the other intermediate traffic states 4, the validation unit 9 labelled B can safely identify this intermediate traffic state 4 as not reliable. As a consequence, this non reliable traffic state 4 is not inputted to the aggregation engine 500 and is therefore not included in the calculation of the traffic state 2 performed by the aggregation engine 500. This drastically increases the reliability of the calculation performed by the aggregation engine 500 as reliable intermediate traffic states 4 are used in the generation of the traffic state 2.

[0054] According to an embodiment shown in Fig. 4, a traffic data fusion system 1 comprises a plurality of interfaces 100, a plurality of data sources 200, a central traffic database 300, a central data fusion engine 400, an aggregation engine 500, a validation unit 7 labelled A and a validation unit 9 labelled B. Traffic data 3 is collected by a plurality of data sources 200. For example in Fig. 4, Floating Car Data source 201, counting loops source 202, travelling time source 203, journalistic data source 204 and/or any other suitable data source 205 receive and collect traffic data 3 for a network of roads. Interfaces 100 receive the traffic data 3 from the plurality of data sources 200. As visible in Fig. 4, each interface 100 receives traffic data 3 from one data source 200. In Fig. 4, interface 101 to interface 105 respectively receive traffic data 3 from data source 201 to data source 205. According to an alternative embodiment, each interface 100 receives traffic data 3 from more than one data source 200. The central traffic database 300 collects the traffic data

3 from the interfaces 100. The central traffic database 300 comprises a road network description 310. A road network description 310 consists of a consistent logical description of the location, properties and topological model of the road network. It contains a set of nodes and links with additional properties described in segments. A node describes a geographical point with coordinates, for example a longitude and a latitude, in a world surface coordinate system. It contains a node ID and additional node properties. Roads are described as links, a vector or connection between different nodes. Links have driving properties like driving direction, number of lanes, speed limits, road length, etc. A set of links and nodes describes a graph that can be used in a route planner or a Geographic Information System. A link can be split in different short sections, referred to as segments. According to an alternative embodiment, the central traffic database 300 comprises the interfaces 100. The central data fusion engine 400 uses the traffic data 3 to generate intermediate traffic states 4. The central data fusion engine 400 performs a plurality of traffic state determination algorithms 10 using the traffic data 3 to independently generate respective intermediate traffic states 4 for the network of roads. The traffic state determination algorithms 10 visible in Fig. 4 are for example model based, neural network, Treiber-Helbing filter, a semantic web approach, a statistical approach, etc. The aggregation engine 500 receives the intermediate traffic states 4 generated by the central data fusion engine 400 and combines the intermediate traffic states 4 to generate the traffic state 2. As visible in Fig. 4, the system 1 further comprises a validation unit 7 labelled A. The validation unit 7 labelled A is adapted to compare the published traffic state 2 with the reality of the traffic state depicted by the traffic data 3 collected by the data sources 200. The validation unit 7 labelled A therefore receives the traffic state 2 and the data 3 as inputs. The validation unit 7 labelled A outputs a comparison 8 of how reliable the published traffic state 2 is compared to the traffic data 3 collected by the data sources 200. The aggregation engine 500 receives the comparison 8 generated by the validation unit 7 labelled A. The aggregation engine 500 takes this comparison 8 into account when determining the weights of the different traffic state determination algorithms 10 in order to alter the reliability of the data sources 200. This way, predictions of the traffic state 2 generated by the traffic data fusion system 1 can be validated by the validation unit 7 labelled A. The weights of one or more traffic state determination algorithms 10 can be adapted so that the traffic state 2 is more in accordance with the traffic data 3. Alternatively, the comparison 8 generated by the validation unit 7 labelled A is used to allocate weights to the traffic data 3. These weights are a measure of the reliability of each of the data sources 200. For example, if the validation unit 7 labelled A identifies that the traffic state 2 does not have any correspondence with the traffic data 3 collected by a data source 200, the validation unit 7 labelled A will generate a weight for the data source 200 smaller than the weights associated with other data sources 200. This way, the validation unit 7 labelled A ensures the traffic data 2 is more in accordance with the traffic data 3 by allocating more importance to one or more data sources 200, i.e. by allocating weights demonstrating a higher numerical value to data sources 200 from which the traffic data 3 is in accordance with the traffic state 2 and/or by respectively allocating weights demonstrating a lower numerical value to data sources 200 from which the traffic data 3 is not in accordance with the traffic state 2. As visible in Fig. 4, the system 1 further comprises a validation unit 9 labelled B. The validation unit 9 labelled B is adapted to compare the intermediate traffic states 4 generated by the central data fusion engine 400 and to delete intermediate traffic states 4 the validation unit 9 labelled B identifies as not reliable. The validation unit 9 labelled B therefore receives the intermediate traffic states 4 as input. The validation unit 9 labelled B outputs reliable intermediate traffic states 4. For example, if one intermediate traffic state 4 represents an intermediate traffic state that is completely different, for example a complete opposite representation and/or prediction than the one of the other intermediate traffic states 4, the validation unit 9 labelled B can safely identify this intermediate traffic state 4 as not reliable. As a consequence, this non reliable traffic state 4 is not inputted to the aggregation engine 500 and is therefore not included in the calculation of the traffic state 2 performed by the aggregation engine 500. This drastically increases the reliability of the calculation performed by the aggregation engine 500 as reliable intermediate traffic states 4 are used in the generation of the traffic state 2.

[0055] According to an embodiment shown in Fig. 5, a traffic state 2 is generated from a plurality of intermediate traffic states 4, labelled ITS_a to ITS_m, where ITS stands for Intermediate Traffic State and where m is an integer higher than 1. According to an alternative embodiment, a traffic state 2 is generated from two or more intermediate traffic states 4. The intermediate traffic states 4 are weighted by respective and corresponding weights 5, labelled w_a to w_m. As visible in Fig. 3, each intermediate traffic state 4 is indeed weighted by a weight 5. The weights 5 are equal to an integer or a floating number, and can for example for comprised between 0 and 1, 0 and 100, etc. The numerical value of the weights 5 are different. According to an alternative embodiment, two or more weights 5 can be equal to the same numerical value. For example, all the weights 5 can be equal to the same numerical value and therefore all the intermediate traffic states 4 have the same importance during the generation of the traffic state 2. The weighted intermediate traffic states 6 are then combined together and the traffic state 2 is generated from the combination of at least two intermediate traffic states 4. For example, the weights 5 associated with all the intermediate traffic states 4 except two intermediate traffic states 4 can be equal to zero. In that case, the traffic state 2 is generated from the combination of two intermediate traffic states 4. Depending on the fact that the numerical value of the weight 5 associated with an intermediate traffic state 4 is higher or lower than the numerical value of the weight 5 associated with another intermediate traffic state 4, the intermediate traffic state 4 respectively has more or less importance during the generation of the traffic state 2 than the other intermediate traffic state 4.

[0056] Fig. 6 shows a suitable computing system 800 for hosting the system 1 of Fig. 1. Computing system 800 may in general be formed as a suitable general purpose computer and comprise a bus 510, a processor 502, a local memory 504, one or more optional input interfaces 514, one or more optional output interfaces 516 a communication interface 512, a storage element interface 506 and one or more storage elements 508. Bus 510 may comprise one or more conductors that permit communication among the components of the computing system. Processor 502 may include any type of conventional processor or microprocessor that interprets and executes programming instructions. Local memory 504 may include a random access memory (RAM) or another type of dynamic storage device that stores information and instructions for execution by processor 502 and/or a read only memory (ROM) or another type of static storage device that stores static information and instructions for use by processor 504. Input interface 514 may comprise one or more conventional mechanisms that permit an operator to input information to the computing device 800, such as a keyboard 520, a mouse 530, a pen, voice recognition and/or biometric mechanisms, etc. Output interface 516 may comprise one or more conventional mechanisms that output information to the operator, such as a display 540, a printer 550, a speaker, etc. Communication interface 512 may comprise any transceiver-like mechanism such as for example two 1 Gb Ethernet interfaces that enables computing system 800 to communicate with other devices and/or systems, for example mechanisms for communicating with one or more other computing systems 900. The communication interface 512 of computing system 800 may be connected to such another computing system by means of a local area network (LAN) or a wide area network (WAN, such as for example the internet, in which case the other computing system 580 may for example comprise a suitable web server. Storage element interface 506 may comprise a storage interface such as for example a Serial Advanced Technology Attachment (SATA) interface or a Small Computer System Interface (SCSI) for connecting bus 510 to one or more storage elements 508, such as one or more local disks, for example 1TB SATA disk drives, and control the reading and writing of data to and/or from these storage elements 508. Although the storage elements 508 above is described as a local disk, in general any other suitable computer-readable media such as a removable magnetic disk, optical storage media such as a CD or DVD, -ROM disk, solid state drives, flash memory cards, ... could be used. The system 800 described above can also run as a Virtual Machine above the physical hardware.

[0057] The system 1 of Fig. 1 can be implemented as programming instructions stored in local memory 504 of the computing system 800 for execution by its processor 502. Alternatively system 1 of Fig. 1 could be stored on the storage element 508 or be accessible from another computing system 900 through the communication interface 512.

[0058] Although the present invention has been illustrated by reference to specific embodiments, it will be apparent to those skilled in the art that the invention is not limited to the details of the foregoing illustrative embodiments, and that the present invention may be embodied with various changes and modifications without departing from the scope thereof. The present embodiments are therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description, and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein. In other words, it is contemplated to cover any and all modifications, variations or equivalents that fall within the scope of the basic underlying principles and whose essential attributes are claimed in this patent application. It will furthermore be understood by the reader of this patent application that the words "comprising" or "comprise" do not exclude other elements or steps, that the words "a" or "an" do not exclude a plurality, and that a single element, such as a computer system, a processor, or another integrated unit may fulfil the functions of several means recited in the claims. Any reference signs in the claims shall not be construed as limiting the respective claims concerned. The terms "first", "second", third", "a", "b", "c", and the like, when used in the description or in the claims are introduced to distinguish between similar elements or steps and are not necessarily describing a sequential or chronological order. Similarly, the terms "top", "bottom", "over", "under", and the like are introduced for descriptive purposes and not necessarily to denote relative positions. It is to be understood that the terms so used are interchangeable under appropriate circumstances and embodiments of the invention are capable of operating according to the present invention in other sequences, or in orientations different from the one(s) described or illustrated above.

Claims

1. A traffic data fusion system (1) adapted to produce a traffic state (2) for a network of roads, said traffic data fusion system (1) comprising:

- interfaces (100) adapted to receive traffic data (3) from multiple traffic data sources (200);
- a central traffic database (300) adapted to col-

lect said traffic data (3) from said multiple traffic data sources (200);
- a central data fusion engine (400) adapted to execute a plurality of traffic state determination algorithms (10) using said traffic data (3) to independently generate respective intermediate traffic states (4) for said network of roads; and
- an aggregation engine (500) adapted to combine said intermediate traffic states (4) to thereby generate said traffic state (2).

2. A traffic data fusion system (1) according to claim 1, wherein said plurality of traffic state determination algorithms (10) comprise two or more of:

- a Kalman filter, preferably the ensemble Kalman filter or EnKF;
- a Treiber-Helbing filter, preferably the Extended Generalized Treiber-Helbing filter or EGTF;
- a neural network, preferably a Multi-Layer Feed-forward neural network;
- machine learning;
- pattern recognition;
- a statistic algorithm;
- a genetic algorithm;
- a Lagrange and Euler Cinematic wave method;
- a cumulative number of vehicles;
- a multicriteria method;
- an adaptive smoothing;
- an adaptive Kalman method
- a particle swarm optimization;
- ant colony optimization;
- a dynamic programming; and
- simulated annealing.

3. A traffic data fusion system (1) according to claim 1, wherein said aggregation engine (500) is adapted to calculate a mean value of said intermediate traffic states (4) as said traffic state (2).

4. A traffic data fusion system (1) according to claim 1, wherein said aggregation engine (500) is adapted to calculate a median value of said intermediate traffic states (4) as said traffic state (2).

5. A traffic data fusion system (1) according to claim 1, wherein said aggregation engine (500) is adapted to calculate a weighted sum of said intermediate traffic states (4) as said traffic state (2).

6. A traffic data fusion system (1) according to claim 5, wherein said weighted sum comprises weights (5) representative for the reliability of said respective traffic state determination algorithms (10).

7. A traffic data fusion system (1) according to claim 1, wherein said traffic state (2) comprises one or more of:

- a vehicle volume;
- a vehicle classification;
- a vehicle density;
- an average vehicle speed; and
- journalistic traffic data.

8. A traffic data fusion system (1) according to claim 1, wherein said traffic state (2) is periodically updated.

9. A traffic data fusion system (1) according to claim 1, wherein said traffic state (2) is calculated for road segments of a predetermined length.

10. A traffic data fusion system (1) according to claim 1, wherein said multiple traffic data sources (200) comprise two or more of:

- a Floating Car Data system or FCD system;
- a point measurement system;
- a trajectory measurement system;
- a journalistic data system;
- a weather data system; and
- a traffic management data system.

11. A method for producing a traffic state (2) for a network of roads, said method comprising the steps of:

- receiving traffic data (3) from multiple traffic data sources (200);
- collecting said traffic data (3) from said multiple traffic data sources (200);
- executing a plurality of traffic state determination algorithms (10) using said traffic data (3) to independently generate respective intermediate traffic states (4) for said network of roads; and
- combining said intermediate traffic states (4) to thereby generate a traffic state (2).

12. A computer program comprising software code adapted to perform the method according to claim 11.

13. A computer readable storage medium comprising computer-executable instructions which, when executed by a computing system, perform a method according to claim 11.

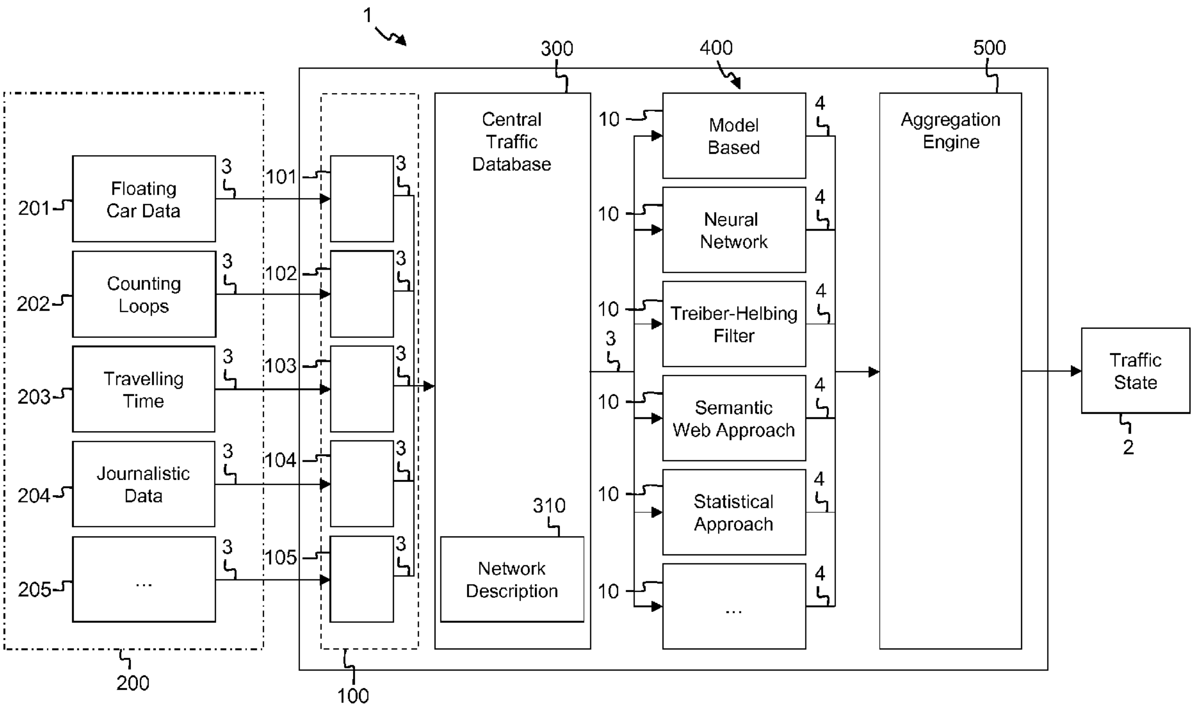

1
300
400
500
Floating Car Data
Counting Loops
Travelling Time
Journalistic Data
...
201
202
203
204
205
200
3
101
102
103
104
105
100
Central Traffic Database
310
Network Description
10
Model Based
Neural Network
Treiber-Helbing Filter
Semantic Web Approach
Statistical Approach
...
4
Aggregation Engine
Traffic State
2

Fig. 1

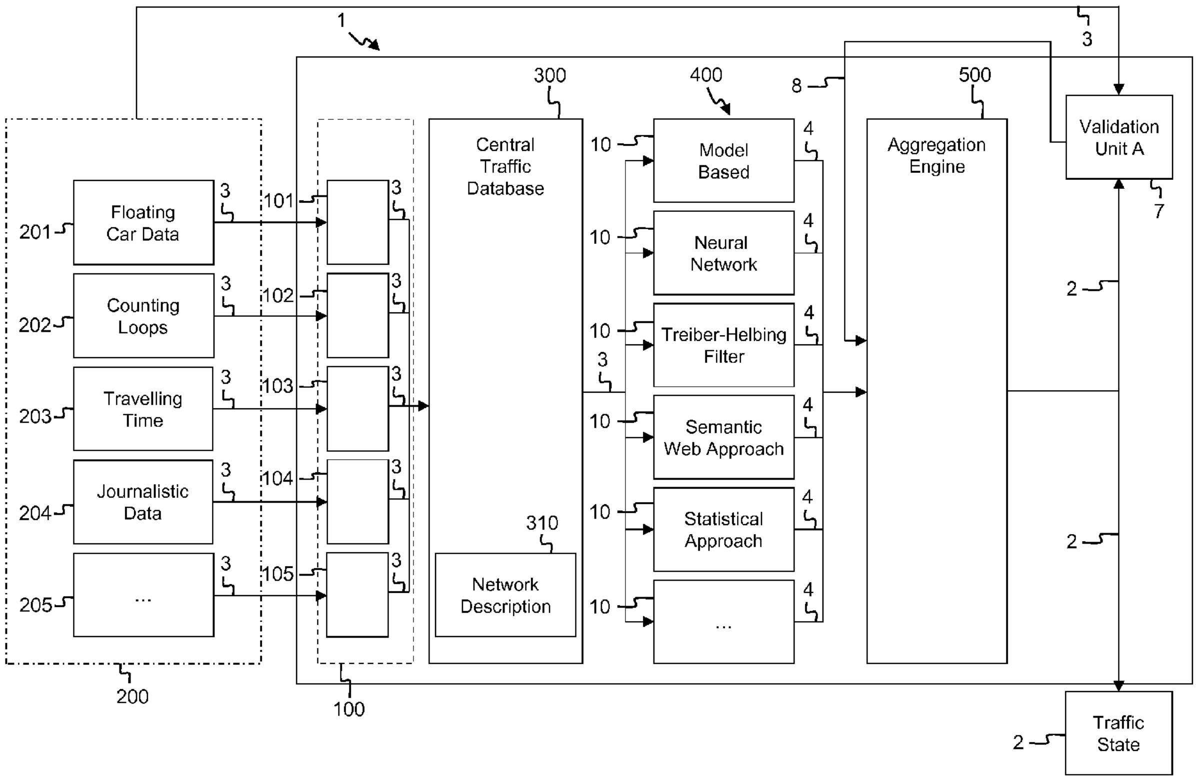

1
300
400
8
500
3
Validation Unit A
7
Central Traffic Database
10
Model Based
4
Neural Network
Treiber-Helbing Filter
Semantic Web Approach
Statistical Approach
...
Aggregation Engine
2
201
Floating Car Data
202
Counting Loops
203
Travelling Time
204
Journalistic Data
205
101
102
103
104
105
310
Network Description
200
100
Traffic State

Fig. 2

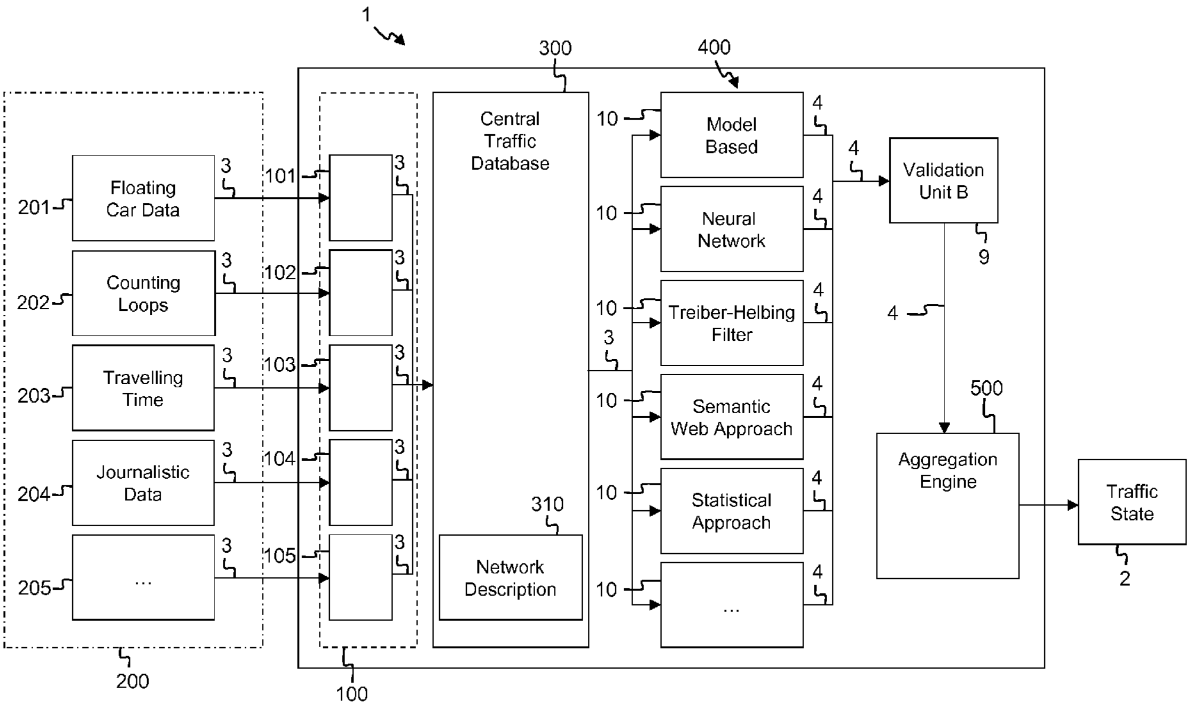
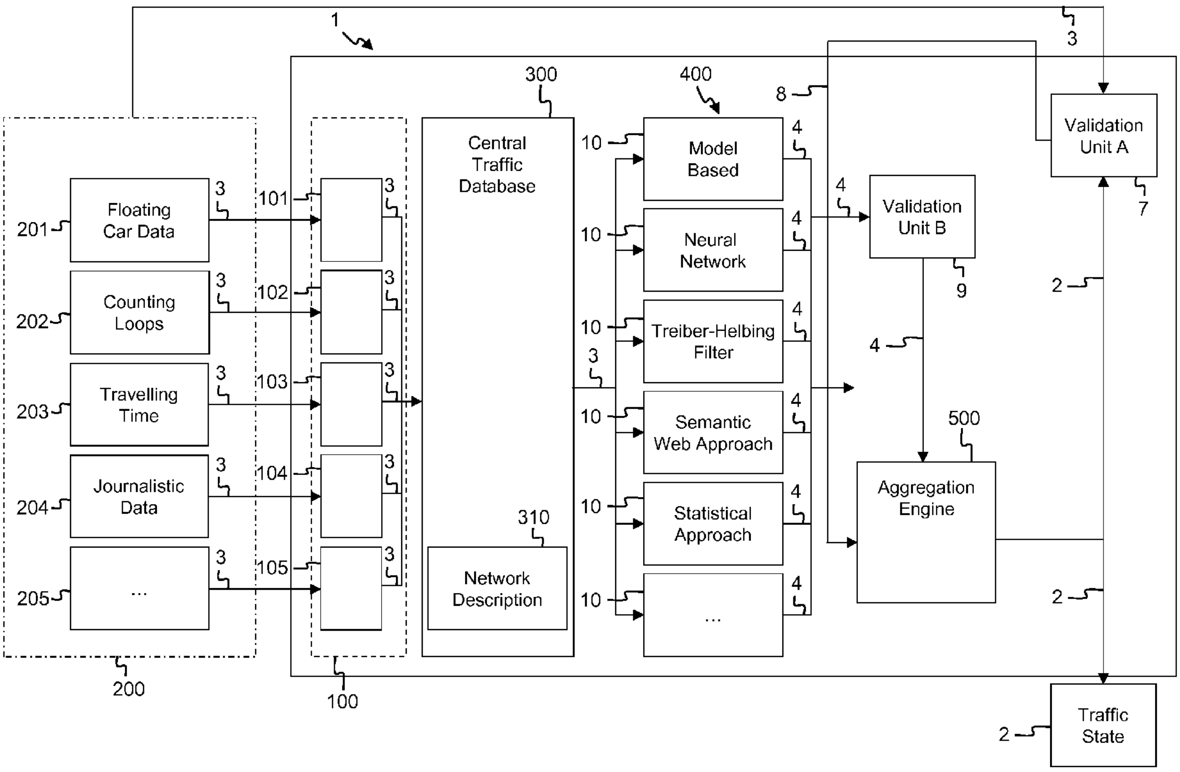

1
300
400
Central
Traffic
Database
Network
Description
310
Floating
Car Data
Counting
Loops
Travelling
Time
Journalistic
Data
...
201
202
203
204
205
200
101
102
103
104
105
100
3
10
4
Model
Based
Neural
Network
Treiber-Helbing
Filter
Semantic
Web Approach
Statistical
Approach
...
Validation
Unit B
9
500
Aggregation
Engine
Traffic
State
2

Fig. 3



1
3
300
400
8
Central Traffic Database
Model Based
Neural Network
Treiber-Helbing Filter
Semantic Web Approach
Statistical Approach
...
Validation Unit A
Validation Unit B
Aggregation Engine
500
Floating Car Data
Counting Loops
Travelling Time
Journalistic Data
...
201
202
203
204
205
101
102
103
104
105
10
4
7
9
2
310
Network Description
200
100
Traffic State

Fig. 4

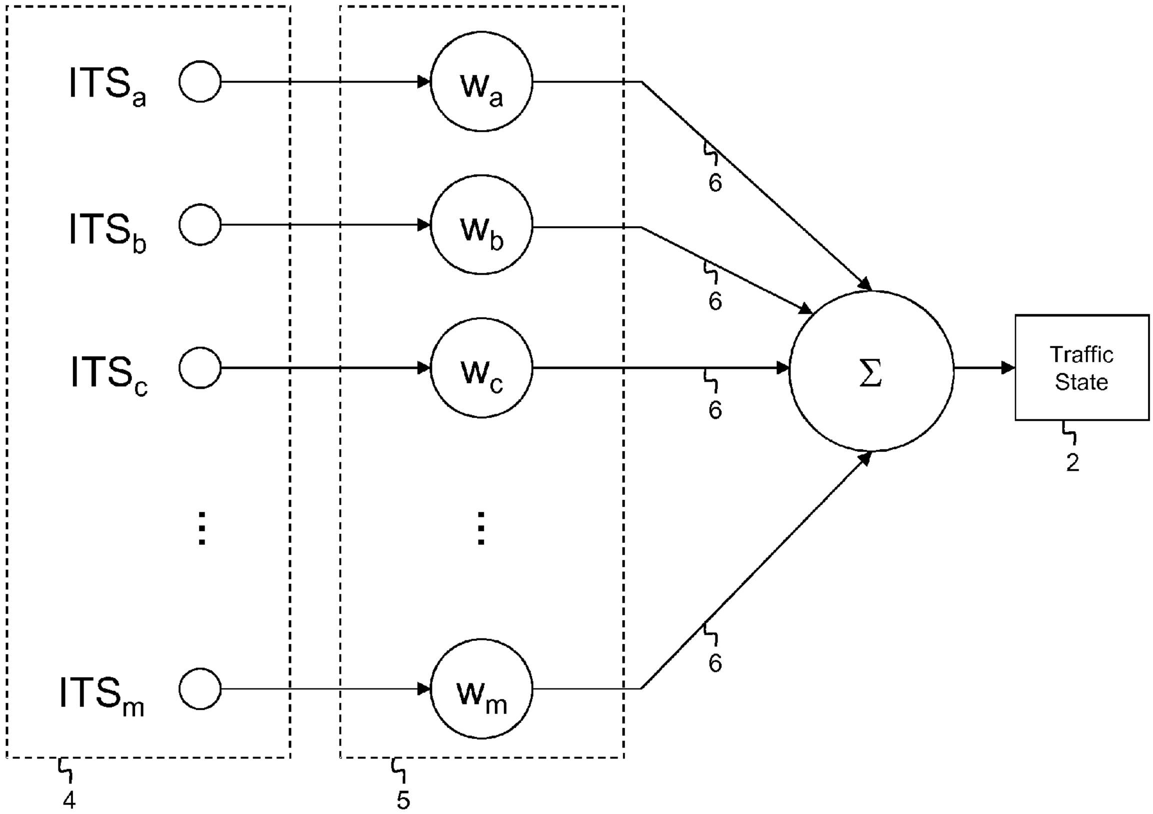


Fig. 5

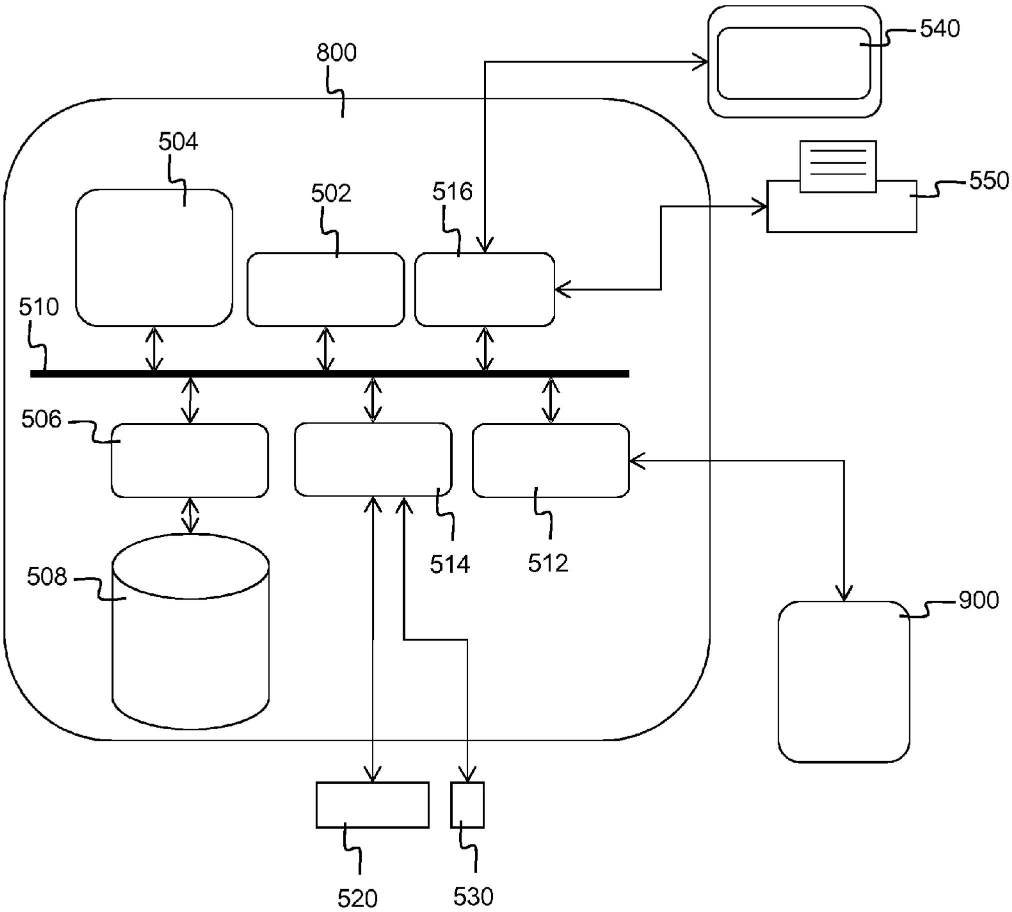


Fig. 6

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 14 19 9054

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 5 173 691 A (SUMNER ROY L [US]) 22 December 1992 (1992-12-22) * column 8, line 62 - column 9, line 15; figure 2 * * column 11, line 62 - column 12, line 4 * * column 12, line 35 - line 64 * -----	1-13	INV. G08G1/01
A	US 2007/118275 A1 (QI HONGWEI [CN] ET AL) 24 May 2007 (2007-05-24) * paragraph [0084]; figure 7 * -----	1-13	
A	US 2005/099322 A1 (WAINFAN S L [US] ET AL WAINFAN S LYNNE [US] ET AL) 12 May 2005 (2005-05-12) * paragraphs [0039] - [0041] * -----	1-13	
A	DE 199 44 891 A1 (MANNESMANN AG [DE]) 20 April 2000 (2000-04-20) * figure 7 * -----	1-13	

TECHNICAL FIELDS SEARCHED (IPC)

G08G

The present search report has been drawn up for all claims

1

Place of search	Date of completion of the search	Examiner
The Hague	21 May 2015	Créchet, Patrick

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 14 19 9054

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

21-05-2015

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5173691 A	22-12-1992	NONE	
US 2007118275 A1	24-05-2007	CN 1967524 A	23-05-2007
		JP 2007141243 A	07-06-2007
		US 2007118275 A1	24-05-2007
US 2005099322 A1	12-05-2005	NONE	
DE 19944891 A1	20-04-2000	AT 258330 T	15-02-2004
		DE 19944888 A1	10-02-2000
		DE 19944889 A1	23-11-2000
		DE 19944890 A1	23-11-2000
		DE 19944891 A1	20-04-2000
		EP 1057156 A1	06-12-2000
		ES 2211066 T3	01-07-2004
		WO 9942971 A1	26-08-1999

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20020026278 A **[0004]**
- US 8437948 B **[0005]**