Reconstruction of car trips from license plate codes reading : a possible method without sophisticated tools

Jean-Pierre Leyvraz, EPFL, Lausanne
Philippe Mattenberger, EPFL, Lausanne
André Robert-Grandpierre, Bureau RGR, Lausanne

Conference paper STRC 2002
Session Models

STRC | 2nd Swiss Transport Research Conference
Monte Verità / Ascona, March 20-22, 2002

Reconstruction of car trips from license plate codes reading: a possible method without sophisticated tools

Dr. Jean-Pierre Leyvraz
ENAC
EPFL
Lausanne

Phone: 021-693 28 10
Fax: 021-693 50 60
e-Mail: jean-pierre.leyvraz@epfl.ch

Abstract

Urbanistic and traffic engineering decisions must be based on a good knowledge of the trips actually done through the concerned agglomeration. It is not only important to know the origin and the destination of these trips, but also the used route. Theorically the route could be deduced from an assignment model, but these models are precisely more reliable if their utilization is based on observed data.

For the car traffic, it is possible to divide the city into zones and to put screenline or cordon survey stations at zone limits. If the license plate code of each vehicle crossing the station can be read together with the crossing time, then sorting all observations concerning a same code by order of time allows to reconstruct the trip. Its origin is the zone upstream of the first crossed station, its destination the zone downstream from the last one and the route is described by the successive crossed stations. When all these trips are put into a database, suited processings can produce a lot of various information.

Devices allowing for automatic plate reading together with the crossing time are available. However while these devices are appropriated for permanent observations, they may be too expensive and too complicated to use for a few observation hours in a city. In some cases, the solution of human observers has been considered more suited.

The problem is that the human observer does not have generally the necessary time to read (and write down) the complete license plate code ; he/she can just read a subset of characters (what is also better for privacy respect). He/she cannot either write down the time beside each vehicle ; one can just expect him/her writing the codes on a new page at regular intervals (e.g. each 5 minutes).

These human limits result in difficulties of several types.

- If 2 stations have been crossed during the same time slice, the order in the trip must be determined.

- If the license plate codes are not completely read, the same subset of characters could correspond to 2 different vehicles.

- The observer can even be mistaken when writing down a code.

An Access program has been developed using data from the city of Fribourg (Switzerland). It uses an algorithm to generate the trips from the observations and to tackle with these difficulties.

The main basic ideas of the algorithm are the following ones.

- Normally, different subsets of characters correspond to different trips, while similar subsets may correspond either to the same trip or to different trips (done by the same vehicle or by 2 vehicles having almost the same license plate code).

- Oriented links are defined between the stations, expressing the feasible successions in a trip (spatial constraints) ; these constraints help to confirm or reject the hypothesis that the same

subset read in 2 stations corresponds to the same trip ; they can also help to determine the order in the trip when both observations have been done in the same time slice.

- These spatial constraints are complemented with temporal constraints ; for each link, a minimal and a maximal time lag are defined for the passing time at the initial and the final stations.

- Exceptionally the spatial and temporal constraints can be relaxed ; for example, if a subset has been read in two not directly linked stations, but not in an inevitable intermediate station, the program can assume that the vehicle really crossed this station, but that its code was ill-noted.

The database contains the necessary basic data, the plates observations and the generated trips. The trips generated by this algorithm have been validated by the field knowledge of a traffic engineer having experience of this city. The program could thus be transfered without problem into other cities.

Keywords

Traffic Engineering – Transport Modelling – 2nd Swiss Transport Research Conference – STRC 2002 – Monte Verità

1. Goal of the study

Urbanism and transportation planning within an agglomeration must be based on a good knowledge of the traffic (and in particular of the road traffic) in order to understand how traffic will react to the suggested modifications. However this knowledge may be only partial and approximate. It is based on:

- rough and direct observations (for example vehicle counts on a road section during a given period);

- indirect information (for example living place and work or study place of the inhabitants, leading to a partial knowledge of their trip needs);

- behavioural models helping to make up for missing information.

Even if they are not enough to know everything about the traffic, direct observations are the raw material without which a real knowledge is impossible. So it is worthy to extract the maximum information from them.

The available results will depend on the way these observations are carried out.

A roadside **survey station** allows to know the corresponding traffic volume; it is useful, but very local information.

If the city is divided into zones with survey stations at entrance and exit points of these zones, one has a **survey station system**; with such a system a more general knowledge is possible ; the traffic concerning each zone can be estimated (except for the internal traffic).

If instead of just counting the vehicles, one also records for each one some attribute allowing to recognize it (for example its license plate code), as well as the passing time, it is possible to track the vehicles. We will thus speak about **tracking observations**. These observations provide at least 2 kinds of useful information:

- the amount of trip demand from each origin to each destination;

- the most selected paths to drive from origin to destination.

A lot of other information would theorically be available, for example times between successive survey stations, even with their variation amongst motorists. But for that, it would be necessary to know with a sufficient precision the passing times.

2. Concrete organization of tracking observations

Many more or less sophisticated ways of organizing tracking observations can be imagined, depending on the available resources and on the frequency of the survey campaigns.

The present method, tested on an application processed in Fribourg (Switzerland) in 1999, is well suited when the campaigns are not too frequent (for example for campaigns realized every fifth year or carried out exceptionally right before deciding on a significant project. In such cases, manual procedures may be prefered; they are less sophisticated than the automatic solutions, but require much less hardware and overheads.

These survey campaigns can be held over several days, but for minimizing the costs, observations take place only at certain times of the day, over limited **periods** of one or more hours. The **traffic type** is known to vary during the day (peak hour, off-peak hour) or during the week (working or non-working day). The selected periods can thus be similar concerning traffic type (study of day-to-day variance) or dissimilar (study of different conditions).

The agglomeration is divided into zones: on the one hand closed zones whose limits can be clearly specified on a map, on the other hand one or more open zones representing the world outside of the study field.

The survey stations are generally set at zone limits. Normally, they should be put at every entrance or output point of the closed zones, except for those where the traffic is not significant.

The survey stations are directional; therefore a two-way road section needs 2 stations.

The survey station system is the same one for every period of the campaign. In other words during the whole campaign, all survey stations are working exactly at the same hours.

Some **operation rules** are fixed for the whole campaign, independently of the period and of the survey station. They must be complied with by all the observers. As these rules depend on the available resources, they may change from a campaign to another. They relate in particular to the following points.

- Information allowing to recognize when two different vehicle observations at two different stations during the same period correspond to the same trip. In fact one can just hope to identify the same **vehicle**, not the same **trip**. It is not exactly the same thing because a vehicle may carry out more than one trip during the period. The most obvious idea for this identification is to record the license plate code. But the survey staff may not have time enough for writing down the complete code (what is also better for privacy respect). Therefore they just note a part of this one, always the same, for example the first 4 characters. There is obviously the risk to confuse two vehicles having just slightly different codes.

- The **precision** for the **vehicle passing time**. One method could be recording these times by pushing on an ad hoc apparatus each time a vehicle is passing; the method would be very precise, but would cause a supplementary stress to the observers with resulting error risks. Therefore in the case of Fribourg the time was just noted to within 5 minutes. More precisely, at each survey station the license plate codes were noted successively, beginning a new paper sheet each time 5 minutes were completed. Both methods can be supported by the program described here because their formal nature is the same: in both cases indeed the "exact" passing time is not given, but just the **time slice** during which the vehicle is passing. However the slice width is one second in the first case and 5 minutes in the second one, with important consequences : in the latter case a vehicle may cross more than a survey station during the same slice, with the difficulty to determine the sequence order.

3. Trips, route and origin-destination matrix

Ideally, a survey station system should allow to know for each period the trips carried out by the various vehicles. But in fact, one can know only trips crossing at least a survey station. Trips remaining inside a zone without crossing any station are ignored.

Moreover, for the recorded trips, one can at most know the succession of the survey stations and not the exact path. When a vehicle crosses successively two survey stations P_A and P_B, if there are several reasonable paths from P_A to P_B, it is not possible to know which was used.

Concerning the trip origin and destination, one can just know the upstream zone for the first survey station and the downstream one for the last station. In case of open zones, this information is not very precise.

From the trips and their routes, one can compute for each route the number of vehicles following it (the route being obviously defined by the successive survey stations). This allows calculation of the trips origin-destination matrix.

All this construction requires the possibility to reconstruct, from observations carried out at different stations by different persons, the trips under the form of successive crossed stations. This reconstruction cannot result from simple sorting operations (observations in a same period sorted firstly by plate license codes for getting the trips, and secondly by time slice for getting the sequencing order in the trip). The following problems make it more complicated

- If during the same time slice the same vehicle is recorded at two different survey stations, a simple sorting does not allow to know which station was crossed first.

- The same recorded code information at different stations does not guarantee that the vehicle was the same, because only a part of the code was recorded and two different vehicles could have nearly equal codes. Even if the vehicle was the same, the observations could relate to different successive trips performed by the same vehicle.

And finally the survey sheets may contain errors. For example, a vehicle could be omitted at a survey station or its code could be ill-recorded. These errors should not be reflected without verification into the trip determination.

4. Program Transec and its data model

Program Transec for tracking observations storage and processing is under the form of a relational Access database with routines written in Access BASIC. Figure 1 shows the various objects in this database with their relations.

Eight out of the 10 tables are used to store input data from program users. The two last are used to store trips. They are necessary, as trips cannot be simply deduced from input data. The database does not contain tables concerning routes (with the number of vehicles per route) or concerning the origin-destination matrix, since this information can be deduced by queries on the data stored in the 10 tables. These tables are described briefly below.

The **general data** table ***tblDonGen*** contains the general description of the survey and the width of the used time slices.

The **traffic type** table ***tblTrafic*** stores the traffic conditions under consideration in the survey (for example peak hour, off-peak hour).

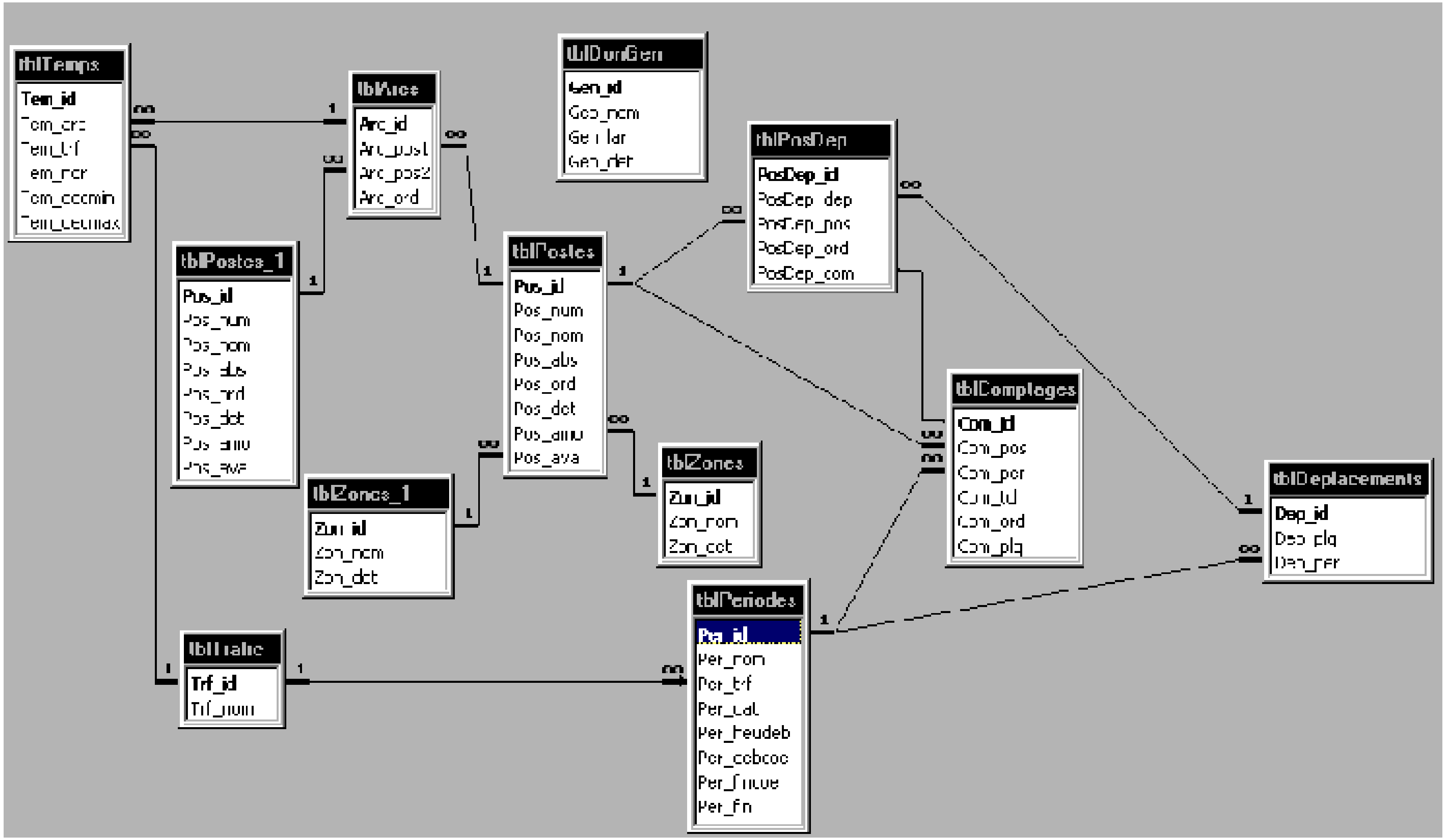
The **period** table ***tblPeriodes*** contains for each period the date, the start and end times, as well as the corresponding traffic type. These periods should not be contiguous in order to prevent a trip extending over 2 periods. The table defines 3 phases during the period: the **initial phase**, the **period core** and the **final phase**. This division is done for avoiding undesirable side-effects. Suppose for example that we are interested in trips between 07:30 and 08:00. A good part of these trips were already begun before 07:30 or will be completed after 08:00. Therefore observations will be carried out from 07:15 to 08:15 and not just from 07:30 to 08:00, in order to contain the whole trip of motorists driving between 07:30 and 08:00.

In such a case, the 3 following phases will be considered:

- the initial phase from 07:15 to 07:30
- the core from 07:30 to 08:00
- the final phase from 08:00 to 08:15.

The **zone** table ***tblZones*** contains information about each of them.

Relations in a Transec database



The **survey station** table ***tblPostes*** contains the identification of each station and gives for each one the upstream and downstream zones.

The **arc** table ***tblArcs*** was introduced for describing the possible successions between survey stations. It is related to a graph whose nodes are the survey stations and whose directed arcs represent the relations between stations. In this graph, an arc P_AP_B means that a vehicle can reasonably cross the survey station P_A, then the station P_B without crossing any intermediate station.

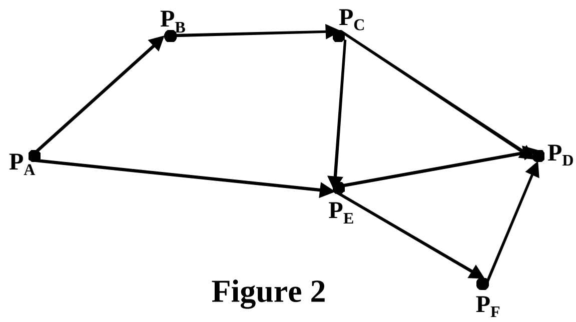


Figure 2

This graph can highlight observation errors. Consider for example figure 2. If a vehicle is detected at P_A and then at P_C, but not at P_B, while about at the same hour another vehicle is detected at P_B and nowhere else, one can imagine the second vehicle being in fact the first one ill-detected.

The **travel time** table ***tblTemps*** contains the travel times on the arcs. For each arc P_AP_B, it contains one record per traffic type. Each record contains the normal travel time from P_A to P_B under the given traffic type. It also contains extreme values under the form of minimum and maximum time lags between the time slice a vehicle is crossing P_A and the time slice the same vehicle is crossing P_B. These time lags are expressed as integer numbers of time slices.

The arcs, with their minimum and maximum time lags under the current traffic type, can be used for checking whether 2 observations at the respective stations P_A and P_B and at the respective time slices T_A and T_B can relate to the same trip.

The **observation** table ***tblComptages*** contains the observations themselves. For each of them are given the survey station, the period, the time slice, the chronological order in its time slice and the part of the license plate code that is recorded in the survey (from now called **abbreviated code**).

The **trip** table ***tblDeplacements*** contains for each trip the period and the abbreviated code.

The **observations-on-a-trip** table ***tblPosDep*** contains one record for each observation, considered as related to a trip. The record gives the trip, the survey station, the rank of the observation on the trip and the reference to the related record of tblComptages.

The survey station field of tblPosDep could seem useless since the records of tblPosDep are pointing on table tblComptages which contains a survey station field. In fact there would be a redundancy if coherence between the tables tblComptages and tblPosDep was perfect. However it is not so, because the table tblPosDep is built during a process including observation errors correction. So if a trip must necessarily cross a survey station P, but the associated abbreviated code was not found there, the trip reconstruction adds the corresponding record into tblPosDep, but without association to any record of tblComptages.

5. Trip reconstruction starting from the observations

The algorithm used in Transec to reconstruct trips starting from observations is composed of several stages. It successively processes each period, dealing with the different abbreviated codes recorded in the period. For an abbreviated code NP, it carries out successively the processes described further in paragraphs 5.1 to 5.4 in order to get the related trips. Then, having processed every period and every abbreviated code, it carries out on the whole database the processes defined in 5.5 and 5.6.

5.1 Reconstruction initialization

In this first stage, all observations of a same period _ having detected the same abbreviated code NP are considered associated to the same trip. Furthermore, it is necessary to arrange all these observations in the right order into a sequence S.

Observations are obviously arranged by chronological order, in other words by ascending slice time order. A problem occurs when several observations concern the same slice time.

The basic idea for breaking the ties is based on the chronological order of each observation in its own time slice at its own survey station. For example, if code NP was recorded amongst the first ones (resp. the last ones) at the station, the vehicle probably crossed the station in the first (resp. the last) part of the slice time. Of course this assumption might be wrong, for example in case of traffic lights : if a light just upstream from the station is red during the first part of the slice, then even the first crossing vehicles are passing during the last part of the slice. But suppose that the assumption is right; if it is so, the same code NP recorded during the same slice time amongst the first ones at station P_1 and amongst the last ones at station P_2 means that the vehicle has done P_1P_2 rather than P_2P_1.

Formally, the tie-breaking rule between observations *i* having detected the same abbreviated code NP during the same time slice at respective survey stations P_i is expressed as follows. Let N_i the number of vehicles counted at station P_i during the time slice and ρ_i the chronological order of the abbreviated code NP in this time slice. The observations are arranged by ascending order of the ratios $\frac{\rho_i - 0.5}{N_i}$.

This rule is satisfactory as long as vehicles crossing the stations are rather numerous and their distribution over the time is rather regular.

5.2 Corrections based on feasible successions

The sequence given by S can be unfeasible in practice. Therefore feasibility checks will be carried out.

Let A and B 2 observations of S, P_A and P_B the related survey stations and T_A and T_B the related time slices. Observations A and B can be **successive** observations from a same trip only if the following **succession constraints** are simultaneously satisfied.

- There must be in tblArcs an arc P_AP_B. The absence of such an arc would mean that it is not possible to go directly from P_A to P_B without crossing an intermediate survey station.

- The time lag T_B - T_A must be at least equal to the minimum time lag for arc P_AP_B, given the current traffic type. Otherwise a vehicle passing at P_A at time T_A does not have time to reach P_B for time T_B and so both recorded vehicles must be different

- The time lag T_B - T_A should not exceed the maximum time lag for arc P_AP_B, given the current traffic type. Otherwise there would be 2 possibilities: either the vehicles detected in observations A and B are different, or it is the same vehicle, which have stopped between both stations. In both cases, the observations relate to different trips.

The first of these succession constraints is called **spatial constraint** and the 2 others **temporal constraints**.

Stage 5.2 is based on the idea that if 2 successive observations of S do not respect the succession constraints, then:

- either these 2 observations correspond to different trips;

- or they correspond to the same trip, but the right order of the observations is not that given by S; for getting the right order, it is necessary to permute observations carried out during the same time slice.

The applied correction method is roughly speaking the following one.

For getting the first trip related to the current period and abbreviated code, the observations are scanned according to the order defined by S. The first observation of S is introduced as trip start. When successive observations of S are scanned, 3 cases can arise.

- The succession constraints allow to append the observation at the end of the trip being generated; then it is done so.

- They do not allow that, but allow an insertion upstream in the trip; this insertion is thus carried out.

- Both above mentioned possibilities are excluded; the observation will thus not be related to this trip.

If the observations of S could not be all related to a single trip, the operation is repeated for getting the second trip and so on. Obviously, the observations already related to a trip are no longer taken into account.

More formally, the method can be expressed as follows.

The following notations will be used.

n	=	number of observations in sequence S
$P(i)$	=	survey station of i-th observation of S ($1 \leq i \leq n$)
$T(i)$	=	time slice of i-th observation of S ($1 \leq i \leq n$)
$Inclus(i)$		true if the i-th observation of S is already included in a trip
N_d	=	number of trips reconstructed from S
$L(d)$	=	number of observations for the d-th trip ($1 \leq d \leq N_d$)
_(j, d)	=	rank in S of the j-th observation of the d-th trip
$A(i', i'')$		true if the 2 following conditions are met:

1) there is an arc a from $P(i')$ to $P(i'')$

2) the time lag $T(i'')$-$T(i')$ lies between the minimum and maximum allowed time lags on a

1) **Research initialization**

Set $N_d := 0$ and $Inclus(i) := False \quad \forall 1 \le i \le n$

2) **Research completed ?**

Let $E := \{i \mid (1 \le i \le n) and\ (Inclus(i) = False)\}$. If E is empty, stop : the research for sequence S has been completed.

3) **New trip initialization**

Let $k := \underset{i \in E}{Min}\, i$. Set $N_d := N_d + 1$, $L(N_d) := 1$, $\omega\ (1, N_d) := k$ and $Inclus(k) := True$.

4) **Check for the next observation in the sequence**

Set $k := k + 1$. If $k > n$ (no other possible observation for this trip), go to 2).

5) **Verification that the observation is not yet included in a trip**

If $Inclus(k)$, go to 4).

6) **Possibility to append the observation to the trip ?**

If $A(\omega\ (L(N_d), N_d), k)$, then set $L(N_d) := L(N_d) + 1$, $\omega\ (L(N_d), N_d) := k$, $Inclus(k) := True$ and go to 4).

7) **Preliminary test for possibility of upper inserting into the trip**

Si $T(k) > T(\omega\ (L(N_d), N_d))$ (impossible permutation between observations done during different time slices), go to 4).

8) **Research for a possibility of upper inserting into the trip**

Let F the set of indices l simultaneously meeting the following conditions:

i) $1 \le l \le L(N_d)$

ii) $T(\omega(l, N_d)) = T(k)$

iii) $A(k, \omega(l, N_d))$

iv) $l = 1 \; or \; A(\omega(l-1, N_d), k)$

If F is empty (upper insertion not possible), go to 4). Else, let $l^* := \underset{l \in F}{Max}\, l$.

9) **Upper insertion**

For $l := L(N_d)$ down to l^*, set $\omega(l+1, N_d) := \omega(l, N_d)$; then set $\omega(l^*, N_d) := k$, $L(N_d) := L(N_d) + 1$, $Inclus(k) := True$ and go to 4).

5.3 Temporal relaxation of succession constraints

After stage 5.2, observations of S may be either all allotted to the same trip or distributed between several trips. The latter case means that during the same period _ either the same vehicle completed several trips (what is possible, but rather unusual for about one hour periods) or there were 2 vehicles in the network with almost the same license plate code (what is too rather unusual).

One thus expects in most cases a single trip for the abbreviated code NP. Therefore stage 5.3 tries as far as possible to "weld" different trips generated by stage 5.2, using temporal relaxation of succession constraints. This relaxation allows a slight temporal constraints violation, considering that in very particular circumstances an arc time could be slightly shorter or longer than one would reasonably imagine. To this end are introduced a concept of **relaxed maximum time lag** obtained by adding one time slice to the maximum time lag and a concept of **relaxed minimum time lag** obtained by taking away one time slice from the minimum time lag (provided it is not already zero).

With **temporal relaxation**, succession constraints are modified as follows. Let A and B 2 observations of S, P_A and P_B the related survey stations and T_A and T_B the related time slices. For A and B being successive observations related to a same trip, it is necessary that the 3 following conditions are simultaneously satisfied.

- There must be an arc P_AP_B in tblArcs.
- The time lag T_B - T_A must be at least equal to the relaxed minimum time lag for arc P_AP_B under the current traffic type conditions.
- The time lag T_B - T_A must not exceed the relaxed maximum time lag for arc P_AP_B under the current traffic type conditions.

The two last constraints are called **temporal relaxed constraints**.

So when stage 5.2 results in more than one trip for the period _ and the abbreviated code NP, stage 5.3 scans the first and the last observation of each trip. If the first observation of a trip can be the successor of the last observation of another, using temporal relaxation of succession constraints, welding is performed between these trips.

5.4 Spatial relaxation of succession constraints

Just as stage 5.3 tries to weld trips using temporal relaxation, stage 5.4 tries to weld them using spatial relaxation.

Suppose that for going from a survey station P_A to a station P_B it is necessary to run either through station P_C or through station P_D. Suppose that the abbreviated code NP has been detected at P_A and then at P_B, but was found neither in P_C nor in P_D. If the observers were infallible, the logical conclusion would be that two different vehicles were detected in P_A and P_B. However, the staff being subject to mistakes, both observations could relate to the same vehicle gone through P_C or P_D without detection, due to one of the following errors.

- The vehicle was not recorded at P_C (resp. P_D).
- The vehicle was recorded, but with a wrong abbreviated code.

The first error type is far less likely than the second one, especially if the observers were asked to give in any case an abbreviated code, even false, just to ensure correct traffic volumes at the survey stations.

Both error types do not have exactly the same consequences. In the first case, the consequence is only the lack at station P_C (or P_D) of an observation concerning the abbreviated code NP. In the second case, an additional and somewhat compensatory consequence is having one observation in excess at P_C (or P_D) concerning some abbreviated code NP' different from NP. It is difficult to know, amongst the different abbreviated codes recorded at the station, which is NP'. The most reasonable way is assuming that NP' is one of the lonely codes appearing only at this station.

For correcting such errors, Transec introduces the concept of **secondary arc**. A secondary arc is defined between 2 survey stations P_A and P_B if both following conditions are met:

- There is no arc P_AP_B.
- There is a survey station P_C such that arcs P_AP_C and P_CP_B exist.

Should several P_C be possible, the retained station is the one minimizing the sum of normal travel times for P_AP_C and P_CP_B. This station is called the **intermediate station of secondary arc** P_AP_B.

A normal travel time, a minimum time lag and a maximum time lag are defined for the secondary arc by adding the corresponding values for arc P_AP_C and arc P_CP_B. Like for the usual arcs, a relaxed maximum time lag is obtained by adding one time slice to the maximum time lag and a relaxed minimum time lag is obtained by taking away one time slice from the minimum time lag (provided it is not already zero). So relaxed temporal constraints can be defined also for secondary arcs.

When stage 5.3 results in more than one trip for the period _ and the abbreviated code NP, stage 5.4 scans the first and the last observation of each trip. If the first observation B of a trip $\_{}_2$ can be the successor of the last observation A of a trip $\_{}_1$, using a secondary arc of intermediate station P_C (with or without temporal relaxation of succession constraints on this arc), welding is performed between these trips, resulting in a single trip _. In the definition of _ contained in tblPosDep, observation A is followed by a reconstructed observation at P_C, then by observation B.

As the above mentioned observation at P_C has not been really done (or at least not been correctly done), its time slice is not directly known. But since the vehicle was at P_A during slice T_A, it could not reach P_C before a slice T_A^* determined by adding to T_A the relaxed minimum time lag for arc P_AP_C. In the same way, for reaching P_B during the time slice T_B, it could not be at P_C later than the slice T_B^* obtained by taking away from T_B the relaxed minimum time lag for arc P_CP_B.

5.5 Compensation for spatial relaxations

When the stages 5.1 to 5.4 have been performed for every period _ and every abbreviated code NP, the observations artificially introduced into tblPosDep for spatial relaxation purposes must be compensated, if possible.

Let a observation artificially introduced during a period _ at a survey station P_C between time slice T_A^* and time slice T_B^* (see paragraph 5.4). The program is searching the database for a trip with a single observation at station P_C, between the time slices T_A^* and T_B^* of period _. If it at least one such trip is found, one of them is removed from tblPosDep; it does not matter **which** trip is removed : the planners are not interested to the route followed by an individual vehicle, but just to the **number of vehicles** performing each route. Should no such trip be found, compensation is not carried out.

All observations to be compensated are successively processed.

5.6 Elimination of possibly truncated trips

Until here it was not considered that a driver could start its trip before the beginning of the observations or complete it after their end. Should it happen, significant bias could be caused. For example, a vehicle crossing successively survey station P_A before the survey period and station P_B during this period could be recorded as a vehicle just crossing P_B but not P_A, which gives a wrong idea of its route.

Precisely for this reason the concepts of initial phase, period core and final phase were introduced into Transec. If for a trip obtained at the end of stage 5.5 the last station was already reached before the period core, there is a good chance that the vehicle was already running in the survey field before the beginning of period. In the same way, if the first station was crossed only after the end of period core, there is a good chance that the vehicle will still be running in the survey field after the end of period.

For that reason, stage 5.6 removes all trips whose first (resp. last) recorded station was crossed after (resp. before) the period core.

6. Application

A first version of Transec was created for a survey traffic carried out at Lausanne in 1983 (Dubois *et al.*, 1985). The program was written in FORTRAN by Philippe Mattenberger from Swiss Federal Institute of Technology in Lausanne on behalf of Bureau Robert-Grandpierre and Rapp SA. The various experiments carried out with this software allowed it to evolve with time, offering new functionalities and improved algorithms for getting the results.

In 1999, it was decided to build from scratch a completely new version of Transec under the form of an Access database with Access Basic routines. This language change tremendously facilitated the update for both following reasons.

- Since no line from the old source was recovered, the bugs inevitably occurring in case of too daring modifications in a program were avoided.

- The relational model made it possible to easily add the new necessary data and the new desirable processings.

The new version was tested on the case of the lower part of Fribourg. Studies are carried out indeed periodically to determine the through traffic in this area. The data used in the test are those of the 1999 survey.

The problem is small-sized and thus ideal for tests: 6 zones, 10 survey stations and 14 arcs. Only one period was considered in the test, at Tuesday April 27, 1999 from 07:15 to 08:15, the period core extending from 07:30 to 08:00. During this period, all survey stations sumed 949 vehicle observations.

The trip reconstruction algorithm required 3 spatial relaxations. In other words, 3 artificial observations were added in tblPosDep to those recorded by the observers. In 2 cases, the matter was probably an ill-read license plate code, because the addition could be compensated by a suppression. In the third case, the addition could not be compensated: the vehicle was probably missed by the observers.

From all these observations, the program generated 388 trips. Only 580 observations were allotted to one of these trips, the others being related to trips completed during the initial phase or started during the final phase.

The choices carried out by the trip reconstruction algorithm, in particular when an abbreviated code corresponds to more than one trip, were manually examined by Bureau Robert-Grandpierre and Rapp SA on the basis of its field knowledge of Fribourg and found quite plausible.

After this first test, the program will probably be used in a larger-scale application on Lausanne.

7. References

J.-M. Dubois et J.-A. Gonzalez (1985) Enquêtes "origine-destination" à Lausanne en 1983, *strasse und verkehr* **No 3**, 104-109, VSS Schweizerischer Verband des Strassen- und Verkehrsfachleute, Zürich.