AUTOMATIC ROAD EXTRACTION THROUGH LIGHT PROPAGATION SIMULATION

Laurent GUIGUES, Jean-Marc VIGLINO
Institut Géographique National, France
Laboratoire MATIS
Laurent.Guigues@ign.fr, Jean-Marc.Viglino@ign.fr

Technical Commission Session III/4

KEY WORDS: Automatic Road Extraction, Parallel Contours Detection, Dynamic Programming.

ABSTRACT

We present a road extraction system from aerial imagery guided by an imprecise road network. After a general discussion on road extraction strategies, we focus on the road recognition module we propose. It relies on two principal elements : a general purpose parallel structures detector in images, which uses a simulation of light ray propagation and reflections; and a dynamic programming based algorithm which constructs a network of hypothetical roads from the previously detected parallelism. We then use a voting scheme involving numerous criteria to select the most probable run of the road. We show through a set of examples that this system works well in rural areas, and also has promising possibilities for the interpretation of complex situations, especially those of multiple lanes roads, like motorways.

RÉSUMÉ

Nous présentons un système d'extraction de routes à partir d'imagerie aérienne, guidé par un réseau routier imprécis. Après une discussion générale sur les stratégies d'extraction de route, nous focalisons sur le module de reconnaissance de routes que nous proposons. Il s'appuie sur deux éléments principaux : un détecteur des structures parallèles d'une image, fonctionnant par simulation de propagations et de réflexions de rayons lumineux; et un algorithme fonctionnant par programmation dynamique qui construit à partir du parallélisme précedemment détecté un réseau de routes hypothétiques. Un vote impliquant de nombreux critères est alors utilisé pour sélectionner le parcours le plus probablement emprunté par la route. Nous montrons à travers un ensemble d'exemples que ce système fonctionne correctement en zone rurale, et se révèle également prometteur pour l'interprétation de situations complexes, en particulier celles des routes à plusieurs voies, comme les autoroutes.

1 DESIGN OF A ROAD EXTRACTION SYSTEM

The problem we address here is the design of an automatic system to perform the following task :
The system is initially loaded with a imprecise road network and an image of mid to high resolution (<1m). The goal of the application is to output a geometrically corrected network matching the image. The initial data-base (DB) precision has to be known in order to set a search distance in the image. The expected final accuracy is the one of the image, that is we aim at a precision of 1 pixel on the position of the axis of the roads. The system is *not* designed to perform any change detection and thus trusts the initial DB topology, and outputs a corrected network that shares this topology.
We remark that the *global* topological correctness assumption is in general violated : usually, available external data either come from digitised maps (manually or automatically) or from lower scale DBs so that a cartographic generalisation process is deeply rooted in the data. We soon remark that the last remark was *global*, that is that such topological generalisations only occur very sparsely, the major part of the generalisation effects being geometrical. We thus assume here topological correctness (an illustration of a typical error produced is visible in the last example).
Two global strategies have been previously proposed to solve the guided road extraction problem : either match the initial network *as a whole* to the image, or *split* the initial network into a collection of linear objects, use a road extraction algorithm working on single roads, and finally reconstruct the network according to the initial topology. The former approach is useful to perform network registration on low resolution images (>1m). The second approach has been widely used for mid to high resolution images (<1m), like in Bordes (1997) for example. This is the one we use, but with a special attention on the initial splitting strategy, arguing that it has important implications on the final reconstruction quality.

1.1 The Network Problem Dissection

In previous works, the independent road portions considered are delimited by the intersections in the network. The initial graph is simply cut at the crossroads, providing a collection of lines, each one being processed independently and finally reconnected. We argue that this splitting scheme has important drawbacks, in particular for the global quality of the reconstructed network. Consider for example a "T" junction, and an extraction scheme where the three incoming roads are processed independently. Focus on the T's bar reconstruction. If the two independent extractions of the two parts of the bar do not converge, because one of the extractions is erroneous or imprecise, the reconstruction will fail to produce a perceptually continuous T's bar. The easiest way to ensure this continuity is to consider the bar as a single object and to perform its extraction as a whole. In this scheme, no reconstruction stage is needed for the bar, and the third incoming road can be snapped very easily to produce the intersection. This strategy can be applied to any kind of junction, by grouping incoming roads into pairs, leaving free only one road in the case of an even number of converging roads. This grouping principle also facilitates the road extraction itself, as road models can be tested on much longer distances as this will be shown in the following.

We thus propose to pre-process the initial graph in order to split it into *long continuous itineraries*. This approach is related to the proposal of Boichis (1998) who gives a hierarchical strategy for the reconstruction of intersections. Our system produces these itineraries by an iterative grouping process : Consider the list of all chainable road pairs, i.e. roads that end up to a common intersection, and while it is not empty, chain the pair that have the best current continuation quality. The continuation quality involved in this process can be a pure geometrical measure of the regularity of the connection, but can also incorporate attributes of the initial DB, if available. For example, if using the IGN BDCarto® as external data, one gets input roads labelled with their name, nature (motorway, main road, secondary road…), number of lanes, and so on. All these data are useful clues to properly evaluate continuations. We thus define a hierarchy of meaningfulness of the available attributes, which is used to compute a continuation quality : e.g. the best quality is given to road sections with the same name, then to connected sections of the same nature, etc. The last criterion considered is geometrical continuity, which, at last, sorts out all conflicting cases.

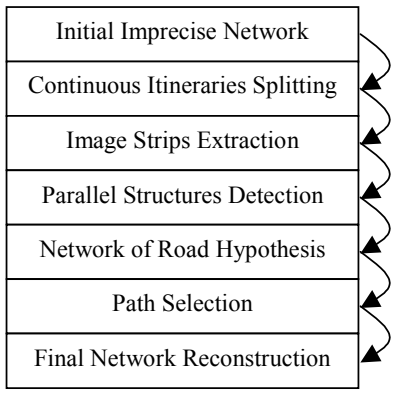


Figure 1. General Architecture of the Road Extraction System

1.2 Linear Road Extraction

Each itinerary is then used independently to extract the corresponding road in the image. A search zone is defined around the initial line and we use a re-sampling technique to turn it into a rectangular strip image. A smoothed version of the line is used to get a regular strip. The width of the strip is set according to the initial DB accuracy. The strip is then obtained by re-sampling the whole image following curvilinear coordinates along the smoothed itinerary. We thus get a rectangular image in which the central line is the trace in the image of the initial itinerary. In a strip image, the true road is expected to run from left to right, connecting the two sides of the image. Typical strip images we obtain are of size 10000x200 pixels. In each strip, a road detection procedure is then used, which outputs a new delineation of the road in strip coordinates. This new road is plunged back in the whole image geometry, and after all strips have been processed, the final network is reconstructed, as discussed before.

In the following, we focus on the road detection procedure used to extract the true run of the road through a strip. Road following techniques have been used for this purpose but they stop when there are obstacles on the road. We rather relate to approaches who try to find road segments hypothesis and connect them, like Hinz, 1999. A first category of approaches to find road hypothesis rely on filtering or morphological approaches, like in Steger, 1998. In our point of view, these approaches use too much constrained models (explicit bar shapes) to cope with the variability of the aspect of the roads. The second category of approaches uses perceptual grouping techniques to find parallel contours. Like them, our approach focuses on the road sides parallelism property and thus use a contour oriented approach. We actually think that parallelism between road sides (and piecewise constant width) is certainly the most stable property of roads. Of course, many other objects have parallel sides : roadsides, rivers, roofs, etc. Yet, following a road for a long distance, even if conflicting parallel-sided objects appear frequently nearby the road, the most continuous strip in the long distance is the road.

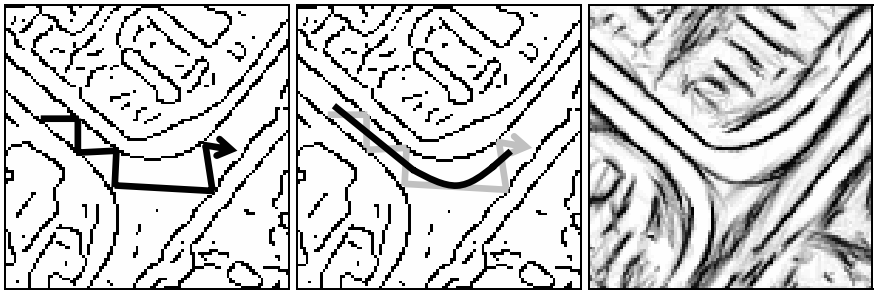
Our road delineation system is articulated in three steps. The first step involves a low-level image processing tool designed to find parallel structures in images. It is described in section 2. The second step constructs a full network of road hypothesis on the basis of the parallelism previously detected. We use a dynamic programming principle described in section 3. Third, the probable run of the road is extracted from the network of hypothesis using a vote involving many criteria that are likely to be satisfied by roads. This is described in section 4. Section 5 finally presents some results obtained.

2 PARALLEL STRUCTURES DETECTION USING A LIGHT PROPAGATION SIMULATION

This section describes the low-level algorithm used to find parallel structures in images. We simply present it in its actual design, avoiding long justifications on the choices we made as it would need significantly much space. It works with the contours extracted from the image, and aims at finding *statistically parallel contours*, possibly noisy, discontinuous or having nested intermediate contours, regardless of their inner grey-level and of the grey-level of the neighbouring objects. Contours are extracted using Canny-Deriche gradient operator, local maxima filtering and thresholding but any other method would suit. The contour image obtained is binary and we don't care any further for the gradient's magnitude. This is motivated by the remark that there is no correlation between the contrast of a line in the image and its likeliness to be a road edge. Thus, after irrelevant contours have been filtered out, we focus on the structures that may be extracted, independently of the initial edges' saliency.

2.1 Estimation of the Shape of two Parallel Curves using the Propagation of a Light Ray

Our algorithm was first inspired by the way a light ray is propagated within an optic-fibber cable. Imagine a ray trapped between two parallel walls, in which it is fully reflected. The ray can be viewed as a sampling process of the two walls. From this sample, one can make an estimation of the complete shape of the guide. The model of parallel structures we use is the class of shapes that can be defined by a medial axis A and a width function w of the curvilinear position on the axis. Then, in the simplest case where the two walls are strictly parallel straight lines, the true estimation of the axis A is obtained by connecting the midpoints of all pairs of successive reflection points : this leads to the medial line of the two parallel walls. The estimation of the width is also straightforward. The point is that these estimations are *independent of the ray considered*, whatever its incidence in the guide may be. It thus produces an estimation scheme independent of any translation, rotation and even scaling factor of the shape. Yet, this midpoint based estimation is biased for curvilinear structures, or structures of non constant width. We thus propose an extension using a piecewise estimation of the curve *(A,w)*. Consider two successive reflections of a ray on two points R_1 and R_2 where the normalised gradient vectors are $\overrightarrow{g_1}$ and $\overrightarrow{g_2}$. The simplest local model that can be proposed is then a circular strip (see fig. 3) of width :



(a) Ray propagation (b) Axis estimation (c) Accumulations

Figure 2. Principle of the Ray Propagation (RP) algorithm

$$w = \frac{\overrightarrow{R_1R_2} \cdot \vec{m}}{2\overrightarrow{g_1} \cdot \vec{m}} \qquad (1)$$

with $\vec{m} = (\overrightarrow{g_1} + \overrightarrow{g_2})/2$. The end points A_1 and A_2 of the axis and its curvature can also be easily computed. Considering then a series of reflections R_i, one gets a piecewise circular model of the shape. Yet, this model is discontinuous. To cope with this problem, we simply average two successive estimations of the position of the same endpoint. The axis A is then modelled by a Cubic Spline passing through the averaged endpoints, with tangent directions constrained to be orthogonal to the gradient vectors. Width is also interpolated in the same way.

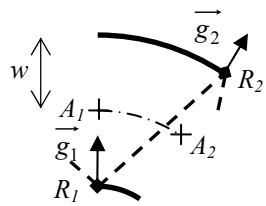
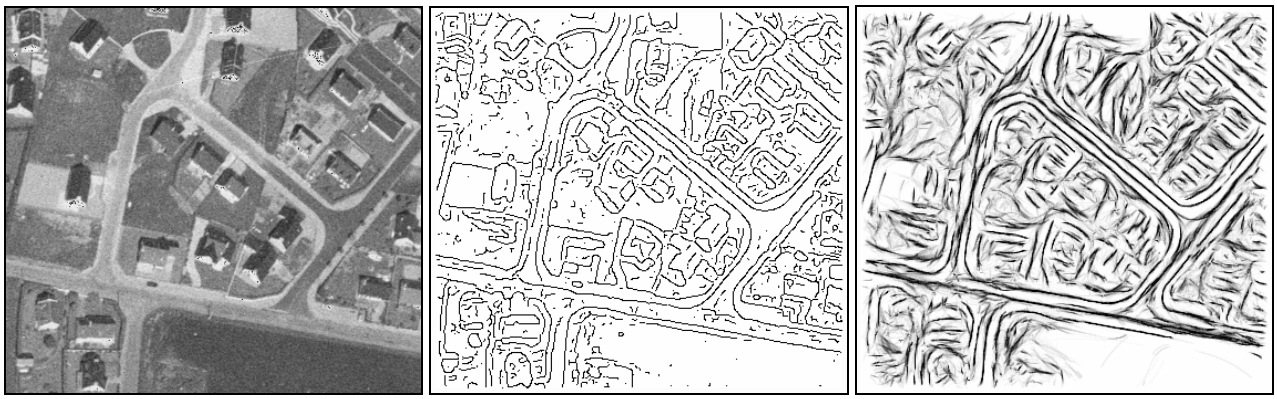


Figure 3. Local circular model estimation.

2.2 Simulation and Accumulation of Estimations

We just described how a single digital light ray can provide an estimation of the shape of two parallel curves. Using this principle, we carry on a simulation of many ray propagation and accumulate the estimations in a Hough Transform like

scheme (Hough, 1962). Each contour point is used as a transmitter from which rays are thrown in a given number of directions. Remark that the digital propagation can be handled very efficiently using Bresenham way of sampling discrete lines. Local model estimations are carried out at each new reflection point and tests are issued to check the validity of the parallel structure that the ray is sampling. The main test concerns the direction of propagation : a ray must always go ahead. Indeed we are not interested in tricky shapes in which a ray might get trapped and rebound in a chaotic way forever. The second test concerns the geometry of the shapes. The detected width and curvature must be within a user-specified range. Third, the piecewise circular model must not be too discontinuous. The aim of this test is to disallow too sparse samples of turbulent shapes, as this would lead to imprecise estimations. This is controlled by bounding the relative variation of two successive width estimations. If any of these tests fails, the ray dies. If it has produced valid estimations for a sufficiently long time (measured with the number of valid reflections done) its estimation is accumulated. The Spline model of the axis is computed as well as the interpolated width function. We use a complex array C for axial accumulations. This array has the same precision than the original image but sub-pixel accuracy could be achieved in a straightforward way. Each point p of the estimated axis contributes to the cell $C(p)$ of the accumulation array in the following way : let $t=e^{ib}$ be the unit vector tangent to the axis at point p, in complex notation. We add it squared to $C(p)$, i.e. $C(p)+=t^2=e^{2ib}$. Squaring t doubles its angle and produces a π-periodic angle, thus adding two such squared numbers leads to a constructive accumulation if the two directions are equal modulo π, and to a destructive accumulation if the two directions are orthogonal. This accumulation principle thus produces interference between axis' directions so that high responses are obtained only when a large number of coherent estimations are made on a same point. At the same time, width and curvature estimations are simply averaged using two dedicated arrays. To handle the presence of noise, creating spurious contours between the lines to extract, transmissions might be allowed. That is, rays might be also forwarded when they meet an obstacle. Without control, this would create a huge binary tree of propagation. This tree is naturally pruned by the geometrical tests issued, and also by limiting the number of successive transmissions allowed. A final control is needed to avoid never ending propagation within loops. We call it the saturation threshold : if more than a given number of estimations have been made at the same point, any ray providing a new one is killed, considering that the local shape have been sufficiently sampled.



(a) Aerial Image (b) Extracted Contours (c) Result of the RP algorithm

Figure 4. Example of parallelism detection with the Ray Propagation algorithm.

An example of parallelism detection using the RP algorithm is given figure 4. Notice that the roads in the original image (4a) have very important variations of aspect. The contours (4b) were extracted setting the Deriche's α parameter to 1, and thresholding the gradient's magnitudes under 15. The result (4c) pictures out the modulus of the accumulated axis directions, that is the parallelism response. Objects of width ranging from 2 to 20 pixels were under scope. We allowed 2 transmissions and a relative variation of width of 0.2. 3 valid reflections were needed to accept an estimation. The saturation threshold was set to 50. As it relies on contours, the algorithm detects parallel structures independently of their grey-level. See for example the dark roofs extracted. Also observe how discontinuous parallel contours as well as very noisy ones are extracted. Finally remark the connections made at the intersections.

3 CONSTRUCTION OF A NETWORK OF HYPOTHETICAL ROADS

The second stage of the detection system uses the outputs of the RP algorithm to find a network of possible roads running from left to right in the strip. Its principle is to find paths along the strip that globally optimise the parallelism response given by the RP algorithm. We consider all the paths connecting the leftmost side of the strip to its rightmost side, and decompose them it into elementary jumps. Each jump is valued by the local parallelism response. The global quality of a path is then defined as the sum of the qualities of its local steps. It is well known that the optimisation problem of finding a path of best global quality can be solved very efficiently by dynamic programming techniques. For

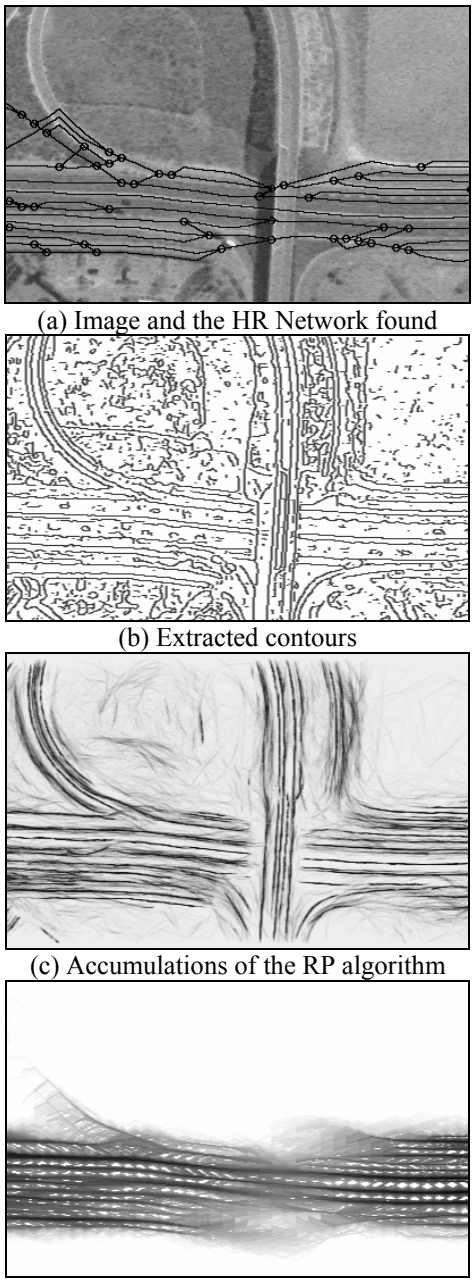
instance, this is the main tool used in stereo-matching problems. It has been also used to find a road connecting two points in an image, see Merlet and Zerubia, 1996. Yet, dynamic programming not only allows to compute the absolute best path, but also provides a way to compute the best path running through *each point of the image*, as well as its quality. This is what we use to construct a complete network of hypothetical roads running the strip, rather than a unique "best" solution.

A regular grid of possible pass points is overlaid on the strip. This grid has a vertical precision of 1 pixel, but is sparse in horizontal direction (we use a 10 pixels jump for images with a resolution of 0.5m). Each point of the grid is connected to two sets of neighbours belonging to the previous and the next column. The vertical range of these connections is defined by setting a maximum angle allowed for a jump. We use 45°, thus each point is connected to 21 left neighbours, and 21 right neighbours. Each jump *(p,q)* is associated a quality $Q_J(p,q)$ computed by averaging the parallelism response over the pixels it passes by. Recall that the axial responses of the RP algorithm are vectors which directions traduce the local direction of the parallel structure. The matching between the direction of the jump and the local direction of the parallel contours detected is introduced in the computation of Q_J by using a scalar product between the two vectors. We denote by $Q_L(p)$ (resp. $Q_R(p)$) the quality of the best path connecting the leftmost side (resp. rightmost side) of the image to a point p of the grid. We call $L(p)$ (resp. $R(p)$) the set of the left neighbours (resp. right neighbours) of a point p. Then the general dynamic programming equation is :

$$Q_L(p) = \max_{q \in L(p)} \{\alpha Q_L(q) + Q_J(q,p)\} \qquad (2)$$

The quality $Q(p)$ of the best path passing through a point p is then given by $Q(p) = Q_L(p) + Q_R(p)$. α is a parameter controlling the weight of the past history of the path. If $\alpha=1$, then all the jumps have equal influence in the total path and optimising from left to right or from right to left leads to the same solution. If $\alpha<1$, the jumps are weighted according to their distance to the point considered : the weight of a jump decreases exponentially with distance. With our data, we use $\alpha=0.8$, so that the weight of a jump is almost zero (<0.01) after 20 steps, that is after 100 *m* in ground units. We also introduce a smoothness term to discard turbulent paths. This is done by adding to equation (n) a third term depending on the angle between the last jump made to reach point q and the current jump *(q,p)*.

Finally, after a forward and a backward dynamic programming optimisation through the strip, we get a *Quality Map (Qmap* in short) $Q(p)$ which gives the quality of the best path passing through each point p. The walk of each path is also stored during dynamic programming by memorising for each point its optimal left an right neighbours, $L^*(p)$ and $R^*(p)$, that is the two neighbours which maximise equation (n). From this *Qmap*, we extract "interesting" paths using the following iterative algorithm :



(a) Image and the HR Network found

(b) Extracted contours

(c) Accumulations of the RP algorithm

(d) *Qmap* of dynamic programming.

Figure 5. Hypothetical Network Construction.

- Extract a set of points called *germs* form the map and sort them following quality order. Each germ is a potential starting point for path extraction. Let L be the sorted list of the selected germs.
- While L is not empty do :
  - Remove the first germ g from L.
  - If g is not on a previously extracted path, nor within an *exclusion distance* d_E from an extracted path :
    - Starting from g, chain the left part of its path using L^* recursively, until the path reaches the left side of the strip or snaps to another previously extracted path. The snap distance to use is d_E.

- Construct the right part of the path using R^* in the same way.

An exclusion distance d_E is used to prevent from extracting too near parallel paths. In our application it is set to 2 pixels, that is 1 ground meter. The way the germs are selected is the key-point of the algorithm. After various attempts using thresholds on qualities, we finally propose a fully qualitative germination criterion based on local optimality of the germs in the map. The first part of the criterion is straightforward : a germ must be a vertical local maximum of the *Qmap*. Local optimality is checked by scanning the columns of the map using a window of width $2*d_E+1$. The second part of the criterion traduces the idea that the paths passing nearby a germ *g* must be consistent, that is they must all elect *g* as a locally unavoidable point. This principle can be expressed by testing the local reciprocity of the connections around a point *p* :

$$R^*(L^*(p))=p \quad \text{and} \quad L^*(R^*(p))=p \tag{3}$$

i.e. the optimal successor of the optimal predecessor of a germ must be the germ itself, in both left an right directions.

An example of the construction of a Hypothetical Roads (HR) network is given figure 5. The *Qmap* image pictured in (d) is constructed by reporting the qualities of the points on every pixel of their optimal left and right jumps. One can clearly see the different locally optimal paths as well as their interconnections. The final HR Network is superimposed to the image in (a). Remark that some paths start to diverge from the main road but break to connect back. This is due to the angular constraint : a jump cannot have an angle greater than 45° with the horizontal. Note that this is an extract from an optimisation carried out on a very much longer strip. The longer example given figure 6 clearly shows that the algorithm constructs the full network of possible tracks in the case of numerous parallel structures.

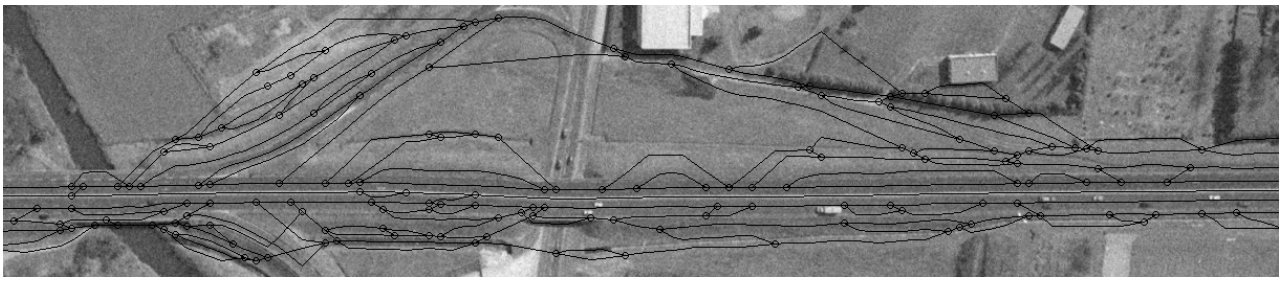


Figure 6. Parallel Structures Network Extraction.

4 ELECTION OF THE FINAL ROAD

In the HR Network constructed, we aim at finding the true run of the road. In the case where the road has a single lane, like in most rural areas, the point is to find the most plausible path in the HR Network, assuming the true road is part of it. First, as the search space has been significantly reduced, more computationally heavy criteria can be used. Second, as road "primitives" have now in a vector structure, including topological connections, high level reasoning can be done. We don't do any "intelligent" reasoning yet, and only consider single lane roads up to now. The only two criteria that have been used in the previous process are road sides parallelism and a smoothness factor in the path extraction. We thus introduce now other unused knowledge concerning road appearance. Roads are likely to be homogenous and of piecewise constant width. To cope with the problem of small obstacles, we introduce a measure of correlation between the histograms of successive road units in the network. Such histograms are made for radiometry and for width of the roads. If the image is coloured, the road should be not too saturated. The low curvature criterion as well as the parallelism response are also reintroduced in this evaluation. Each criterion is then used to find an best path in the HR Network. We thus get the least curved path, the one having the most parallel sides, the most homogenous, the most constant in width, the less colour saturated for colour images, the one which has the best radiometric or width histogram correlation, and so on. A vote is then issued between all these paths to elect a final road.

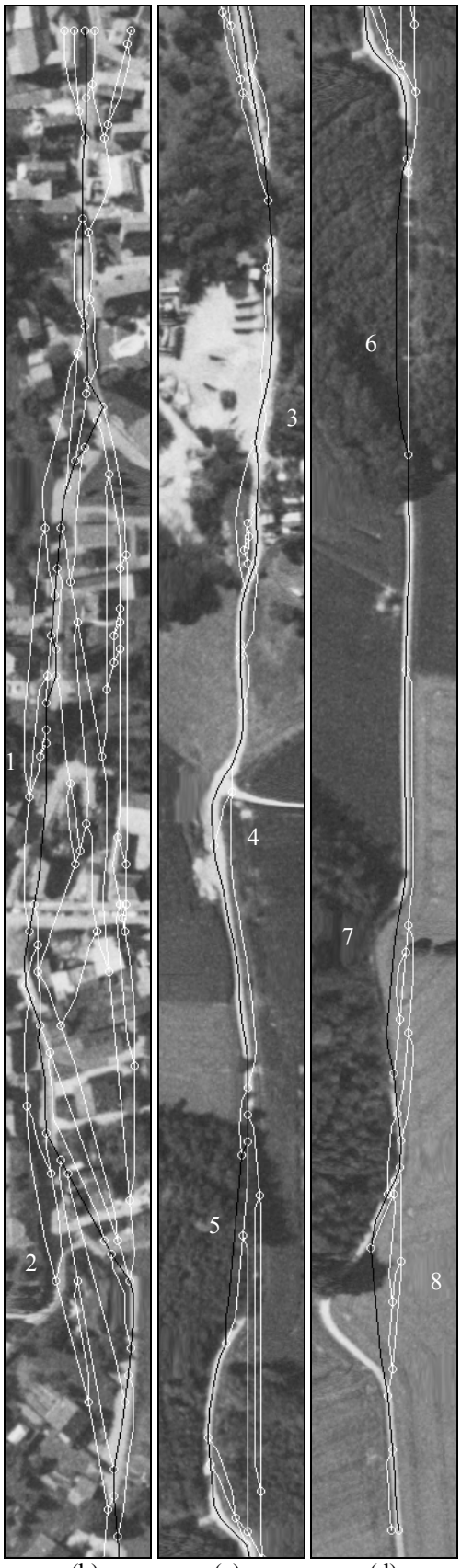
5 RESULTS

Figure 7 gives an example of road extraction in a rural context. 7(a) : 0.5m resolution aerial image with the initial road from IGN BDCarto® pictured in white, and the extracted one in black. 7(b)-(d) : Three parts of the strip image extracted following the prior road from North West to South East. The strip's width is 60m. Recall that in these re-sampled images the trace of the prior road is the central line. The extracted HR Network is superimposed in white and the selected path in black. Various interesting zones are labelled from 1 to 8. Observe first the strip extraction process

which causes geometrical distortions of the original image. It is obvious in central part of 7(c), in zone 4. Yet these distortions generally do not perturb the road extraction algorithm. A problem with the angular constraint on the paths of dynamic programming is visible at point 8 : the angle of the road is more than 45° thus no path can follow the road. The misdetection labelled 7 is due to poor parallelism response in the shadowed part of the road and to the tension factor in dynamic programming. The forest zone 6 shows that the true path was in the HR Network but was not selected certainly because of its darker aspect. Points 2 and 5 show that most times the paths are simply interpolated through long obstacles. Point 3 shows a very interesting behaviour of the parallelism detection algorithm. Whereas one side of the road is almost invisible, it is sufficient to give a response that will be followed. Finally, whereas the village zone 1 is very complex, the algorithm provides an interesting solution relying on very sparse detection.

A global example of road extraction and reconstruction in rural area is given in figure 8 at the end of the paper. The lower left crossroad shows an error induced by the initial DB topological correctness assumption. The white lines are roads that have been evaluated as unreliable by the system.

(a)



(b) (c) (d)

Figure 7. Analysis of the behaviour of our road extraction system in rural area.

6 CONCLUSION

Whereas we present some complete results in rural areas, the main algorithms presented here should rather be considered as first steps toward a full road interpretation system from aerial imagery. The global idea is to produce an hypothetical road network based on objects' sides parallelism, considering this is the most stable property of the roads. This low-level extraction should be the basis for "intelligent" reasoning, involving context understanding, and semantic modelling, like in Hinz, 1999. The most promising development concerns multiple lanes roads recognition with their full structure interpretation.

REFERENCES

Boichis N., Cocquerez J.-P., Airault S., 1998. A Top Down Strategy for Simple Crossroads Automatic Extraction. Proc. of the ISPRS Commission II Symposium, vol. 32, N°2, pp. 19-26.

Bordes G., Cocquerez J.-P., Jamet O., 1997. Road Modeling Based on a Cartographic Database for Aerial Image Interpretation. . Proc. of Semantic Modelling for the Acquisition of Topographic Information from Images and Maps, Birkhaüser Verlag, pp. 123-139.

Hinz S. et al, 1999. Road Extraction in Rural and Urban Areas. Proc. of Semantic Modelling for the Acquisition of Topographic Information from Images and Maps (SMATI), Müchen. Ed. Förstner, Liedtke and Bückner.

Hough P.V.C., 1962. Methods and Means for Recognizing Complex Pattern. US pattent (3069654).

Merlet N. and Zerubia J., 1996. New Prospects in Line Detection by Dynamic Programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(4), pp. 426-431.

Steger C., 1998. An Unbiased Detector of Curvilinear Structures. IEEE Trans. PAMI, Vol. 20, No. 2, pp. 156-169.

Figure 8. A global result in a village.