

Off the Shelf Methods for Robust Portuguese Cadastral Map Analysis

T. Candeias, F. Tomaz, and H. Shahbazkia

Universidade do Algarve – FCT
BIF laboratory, Campus de Gambelas, 8000-Faro, Portugal
{tcandeia, ftomaz, hshah}@ualg.pt

Abstract. A document analysis prototype and its application to the automatic Portuguese cadastral map digitalisation is discussed in this paper. Tuning off the shelf methods and sometimes their extension has permitted to obtain applicable results. These algorithms and their tunings as well as the results obtained are given in the paper. The prototype has been approved for further development to an integrated system to be used by some Portuguese entities.

Keywords: Cadastral Information System, Map Analysis, Image Processing

1 Introduction

In the past years many administrative entities decided to transfer the cadastral information to a numeric format and started using electronic management systems. This is done manually, slowly, and expensively. More than 100000 sheets exist to digitalise in Portugal - one of the smallest countries in Europe.

The process is initiated by a binary scan with a resolution of at least 300dpi. This provides all the necessary information since our map doesn't contain colors or gray scales. The portuguese cadastral map is constrained to several rules, that maintain uniformity explored by our system. Every entity is composed by a closed contour, a numeric identification inserted in a parcel circle, possible dependent plots, separation lines between parcels and a limited word description of each parcel (see figure 1). These are the main guide lines used by our algorithms.

The Portuguese cadastral map authority is pleased with results, that can improve the digitalisation, not only by time saving but also accuracy, since the manual entry of an entire map is tiring and error-prone.

2 Processing Methods

The use of various methods for this analysis are mainly conditioned by the enormous computational effort necessary to complete the task, due to the quantity of information (each map has around 500 entities, 800 parcels, 2500 characters and a significant number of miscellaneous information).

Similar problems were also discussed before [1,2], but to obtain a robust application, all the formal aspects of the Portuguese cadastral maps need to be reconsidered (see section 1).

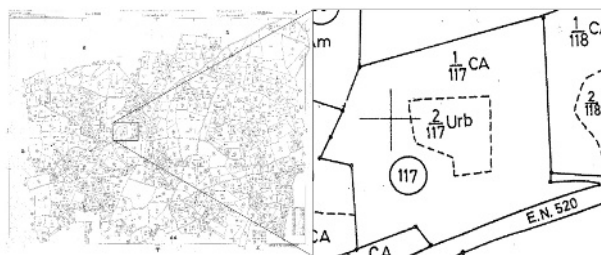


Fig. 1. Overview of a cadastral map sample. This typical map is about 85cm x 75cm representing 138 ha of real land. The smaller portion of the map represents 4cm x 4cm @ 300dpi on the paper map.

The legal status of cadastral administration imposes full robustness therefore the applied methods are largely known, tested and are also necessarily fast and accurate [3].

2.1 Geo-Referenced Crosses

These crosses are present for reference to the real world position of the map. They are present in small numbers, normally well formed (almost no rotation or distortion), well distributed (constant space among them), making it's recognition and extraction very simple to perform, through template matching of perpendicular lines are scaled in function of the image DPI's. It easily obtains results near 100% for recognition, and only a small percentage of the map is damaged while applying our neighbour algorithm for removing.

2.2 Circles

Some problems are associated with circle detection and removal in a common cadastral map: existence of semi-circles, different scales and connection with elements defined as linear (see section 1). This recognition is very important because every parcel is dependent of it's circle recognition.

Hough transform [4,5] is a known process for extraction of parametric defined shapes even in the presence of noise or if the patterns are sparsely digitilised. Some changes were made to the original algorithm, and now we are using a simple, but efficient method similar to the algorithm presented in [6].

This method also covers the recognition of semi-circles, due to the initial element labeling (see section 1). To prevent the necessity of large storing and searching associated with Hough's algorithm, local decision is performed for each space point. This however results in a strong circle mismatch recognition near a "true" circle (see figure 2).

This can be easily solved with a post-treatment searching and confirmation algorithm based on the known characteristics of the map and circles, which in most cases are in conformity with a standard. So the peaks represented in figure 2, can be filtered and chosen through its parameters (distance, radius and power). In the example above, only one circle will be filtered due to the proximity of the other.

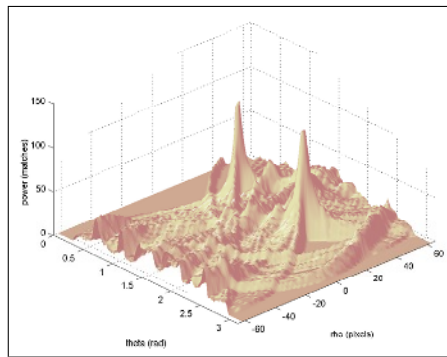


Fig. 2. Hough transform, the local decision accepts 2 circles centers

The results are very good, therefore that proves a working solid algorithm. Indeed, in respect to the cadastral standards, we can recognise correctly, for a general map (containing about 300 circles and 90 semi-circles) about 94% of circles and 90% of semi-circles. The percentage of detected, but inexistent circles are below 0.02% in most maps.

2.3 Dash

The recognition and extraction of dash elements are imperative for the overall result. Similar to the contour of parcels (see section 2.5) the dash represents inner borders of different land applications. The main difficulty is the similarity of the dashes to other components of the map, noise lines or parts of alpha-numeric elements. A real set of dashes can be easily mistaken with other elements from the map due to it's size, shape or context, even to a human's perception.

A density calculation is applied, after a normalization rotation using central moment [7]. In this way, all elements which are not linear and dense are filtered out. A neighbourhood check is also performed to ensure that the dash appears locally in a almost constant frequency.

It's also evident that this method cannot recognise all the dashes (corners or rounded dash), but most important, non-dash elements aren't labelled as a dash by mistake. A pos segmentation is used to join the dash centers and obtain the inner parcel's contour.

2.4 Symbol Recognition

Symbol recognition is important to identify parcel's type, its number and also to detect important out of the analysing zone such as the total number of parcels.

The symbols to be recognised are digits and some characters, which can have different sizes. The small set of characters is due to the limited set of words, that describes parcel's type. After analysing different feature extraction methods [8], it was chosen to implement the zoning algorithm [9].

A database was done for all possible characters which are free of noise. The database can have repeated elements to ensure correct recognition, so it's important to consider the database's consistency. This is obtained only allowing addition of a new symbol if its distance to any other is at least 10%.

The classification is implemented using a distance algorithm between two patterns, which measure their dissimilarities. This method is implemented comparing similar squares in patterns - zoning. These squares are found considering the proportional squares inspecting pattern size. Analyzing and comparing each square a punctuation is obtained that could increase or decrease its match. A big size difference between patterns could also decrease its punctuation.

Using this local matching instead of a global approach, it's possible to obtain an invariant size method.

After applying the distance algorithm between testing symbol within each pattern in database, a list is obtained with all matching percentages. This list is later used to enhance the correct hit rate using a cadastral dictionary.

2.5 Contour Detection

As in section 2.3, the countour extraction gives a list of points which represent the parcel's coordinates. These points are a list of line segments which constitute a closed polyline.

The contour extraction is composed by three stages: detection, correction and vectorization. The contour lines can have two types: continuous or dashed. Each parcel is separated by continuous lines while dashed lines split sub-parcels. The process of detection and correction is different in each case while vectorization is the same.

The precision of extraction is important and must be considered. To increase precision, two neighbour parcels may have the same common segment line. To contribute for robustness of all the extraction process, a contour is represented by its medial axis points [10], otherwise the line segment wouldn't be equal. So the algorithm of contour extraction is applied to the processed medial axis image.

Detection. The main problem of contour detection is the existence of discontinuities which are provided by bad scanning or by noise effect. Two strategies are possible, restore discontinuities or use algorithms which aren't sensible to it. To restore the line following algorithms [11] could be used. But using this kind of algorithm other problems also appear, for example when there are interceptions of lines, the question is which one to follow? Such type of solution would give decision problems that slow down the process. In this way authors have decided to use algorithms which are insensible to discontinuities.

Initially active contour models [12] were considered, but due to the lack of different energy field, it wasn't possible to apply. As every parcel contains only one circle and this information is reliable, the creation of an algorithm with no sensibility to discontinuities and initial knowledge of a point inside the contour made sense.

The first attempt to solve this problem is a fill algorithm, starting in a point inside of a parcel and then coming to all *critical squares*¹ of the contour while filling square after square.

This is the perfect case which represents the application in an image that was every contour closed. But this isn't the case, so instead of applying a normal fill, it's applied a fill with blocks reaching insensibility to discontinuities. This means that discontinuities which are smaller or equal to a block square are the perfect case while others can cause problems. Of course that block size is variable but increasing or decreasing it can cause problems in any parcels of the map.

Another approach is to use a quad tree based algorithm for the parcel segmentation. The main idea is similar to the fill algorithm but now squares are of different sizes. The algorithm starts by finding the biggest square inside a parcel in such a way that no collisions exists with black pixels.

The biggest square is filled and, in an interactive way a 4-neighbourhood expansion occurs with the same squares size. Each new empty square is filled until there aren't more possible empty square with the same size. Then for each non empty squares a split is applied in four equal sizes, and this process continues interactively.

This improving is important because it gives sensibility to a square size and discontinuities can be detected because it's possible to know square size at each interaction. The algorithm stops when a larger square, than the current processing one, is detected and empty of contour pixels.

After the detection of critical squares is complete, a detection to contour points is done. This is implemented considering the filled neighbours squares and detecting all contour points in the opposite direction. After all these procedures are complete, a list of points is obtained for each parcel that represents its contour.

The dash's lines which also represent a contour line are detected considering each element already recognised (see section 2.3).

Correction. After detecting the contour points some discontinuities are found. These discontinuities are due to the occlusions already existent and so, are necessary to correct.

The contour line of continuous type is corrected using a line follow algorithm [11], making a linear interpolation when there are no neighbour pixels to follow.

The dashed line is corrected in a more complex manner. The extrems of the detected dashes are introduced in a list, then a linear interpolation is done to the near extrems, that can't make part of the current dash. After uniting all the dashed line components, the contour that is united to the dashes can be erased. This erosion is done detecting the extremes that didn't interpolate (absolute extremes).

Vectorization. Before applying the vectorization method described in [13,14] its necessary to convert the contour points into chain code.

Once the contour is made by line segments, the chosen method was Rosin & West, considering the split-and-merge approach.

An example of contour extraction can be seen in figure 3.

¹ The critical squares in this case are squares which have black pixels, so there can be known that these are squares with contour points.

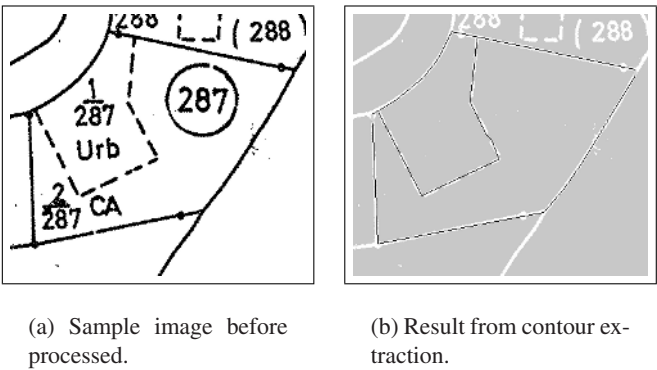


Fig. 3. Example of the extraction process of a parcel's contours.

3 Results

The results were obtained testing each module with a binary cadastral map, 10784x7853 pixels @ 300 dpi. The time was measured with C internal language's function in a Athlon 1400 processor, 512 Mb DDRam and a HD 7.200 rpm.

The program was compiled with gcc version 3.2 (Mandrake Linux 9.0 3.2-1mdk) with CFLAGS = -O2 -march=i686 -mcpu=i686 -funroll-all-loops.

The results can be observed in table 1.

Table 1. Experimental results of application's modules.

Element	Number of patterns	Performance rate	Processing time
Crosse	35	100%	2:23 min
Circle	386	94%	10:16 min
Semi-circle	94	90%	14:32 min
Symbol	4657	77%	12:56 min
Dashe	2347	84%	4:26 min
Contour	450	82%	30:54 min

The patterns were first manually and then automatically classified to obtain the hit rates. Contour results are qualitative, so each vectorized parcell was compared to the initial map.

Global recognition requires 1 hour and 14 min.

4 Discussion

The results obtained at the moment are satisfactory, but can still be improved. In every analysis there are some problems which are resolved increasing consequently the level of the overall results.

The circle detection is only problematic when processing small circles. This happens because the size of non recognised circles are out of the processing range. This can be solved increasing the processing interval but it will be more time consuming.

The semi-circle detection needs the Hough transform to restore lines, which is computationally expensive. Some problems occur in restoring a parcel's contour related to the size of the Hough transform window.

Symbol detection is the most problematic because there are many different classes to classify. The problem is globally solved because each symbol can be classified alone in a correct way. Problems occur when symbols are connected.

The dash detection is also globally solved, but continue problematic when connected to symbols. This may be solved by splitting unknown elements as in the case of symbols.

The contour extraction could be problematic due to previous processing. Other problems occur when there are two closed discontinuities which are removed after applying a linear element detection. This could be solved applying a filter using size and central moments to non-linear elements to restore them close to the contour, in this way long discontinuities are removed.

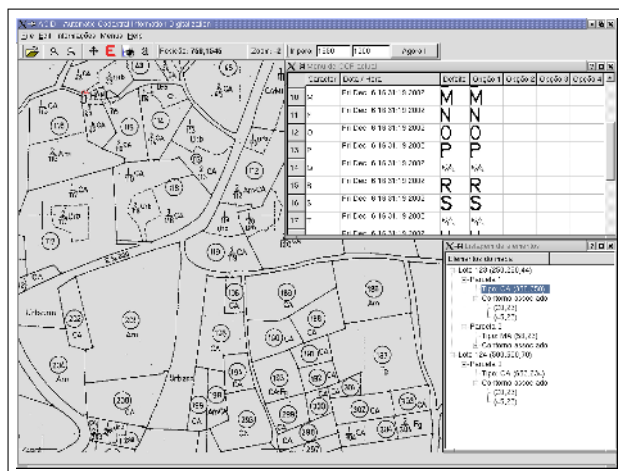


Fig. 4. Snapshot of the prototype running

5 Conclusions

A full assessment of the results only can be carried out when the system goes is operational. But the simplicity of the prototype and the methods used have proven their efficiency in solving the given problem. The use of the meta-knowledge from the beginning of the analysis makes the system dedicated to the task. In general we can consider an automatic digitalisation system as a success if it comes to extract 80% of the data with total robustness. Nevertheless low-level problems such as connected component labelling, cadastral-section boundary extraction and road labelling as well as high-level

problems such as semantic consistency and linking of sections remain a hard task to be addressed. A snapshot of the prototype can be seen in picture 4.

Acknowledgments. This work is funded by the FCT (Fundação para a Ciência e Tecnologia), project ACID, contract SRI/34257/99-00 – Automatic Cadastral Information Digitalization.

References

1. Shahbazkia, H.: Reconnaissance invariante et acquisition de connaissance: application au traitement automatique des plans de cadastre français. PhD thesis, Université Louis Pasteur de Strasbourg (1998)
2. Wenyin, L., Dori, D.: A proposed scheme for performance evaluation of graphics/text separation algorithms. In: GREC. (1997) 359–371
3. Tomber, K., Ah-Soon, C., Dosch, P., Habed, A., Masini, G.: Stable, Robust and Off-the-Shelf Methods for Graphics Recognition. in Proceedings of the 14th International Conference on Pattern Recognition, Brisbane (Australia) (1998) 406–408
4. Duda, R., Hart, P.: Use of the hough transform to detect lines and curves in pictures. Communications of the ACM **15** (1972) 11–15
5. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition **13** (1981) 111–122
6. Matas, J., Galambos, C., Kittler, J.: Progressive probabilistic hough transform. Technical report, University of Surrey/Czech Technical University (1998)
7. Kilian, J.: Simple image analysis by moments. Technical report, Freely distributable, version 0.2 (2001)
8. Trier, O., Jain, A., Taxt, T.: Feature extraction methods for character recognition – a survey (1996)
9. Dimauro, G., Impedovo, S., Pirlo, G., Salzo, A.: Zoning design for handwritten numeral recognition. In: ICIAP (2). (1997) 592–599
10. Tomber, K., Tabbone, S.: Vectorization in graphics recognition: To thin or not to thin. Proceedings of 15th International Conference on Pattern Recognition, Barcelona (Spain) **2**, pages 91–96 (September 2000)
11. Jennings, C.: Computer vision for line drawings. Msc Thesis (1993)
12. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. In: Proc. of IEEE Conference on Computer Vision, London, England (1987) 259–268
13. Tomber, K., Ah-Soon, C., et al.: Stable and robust vectorization: How to make the right choices. Proceedings of Third IAPR International Workshop on Graphics Recognition (Jaipur, India), pages 3–16 (September 1999)
14. Eikvil, L., Aas, K., Koren, H.: Tools for interactive map conversion and vectorization. International Conference on Document Analysis and Recognition **2** (August 14–16, 1995)