

Advances in Portuguese Geographical Document Analysis

H. Shahbazkia, T. Candeias, F. Tomaz, R. Oliveira.

Universidade do Algarve – UCEH

BIF Laboratory, Campus de Gambelas

8000-810 Faro, Portugal

{hshah, tcandeia, ftomaz, rsolivei}@ualg.pt

Abstract – This paper introduces the project ACID (Automatic Cadastral Information Digitalisation), which started in February 2000 and which is financed by the Portuguese FCT (Fundação para a Ciência e Tecnologia). The main goal is to develop a system for automatic digitalisation of Portuguese cadastral maps by using simple but robust image processing algorithms. This paper presents some project goals, and the respective results obtained.

1. INTRODUCTION

ACID is the acronym of the project *Automatic Cadastral Information Digitalisation*. The word *Digitalisation* refers to a transfer of the visual information from a paper support to a Geographical Information System (GIS). Every city council in Portugal has to manage its cadastral information. This information is usually stored on paper support, like the one showed in figure 1.

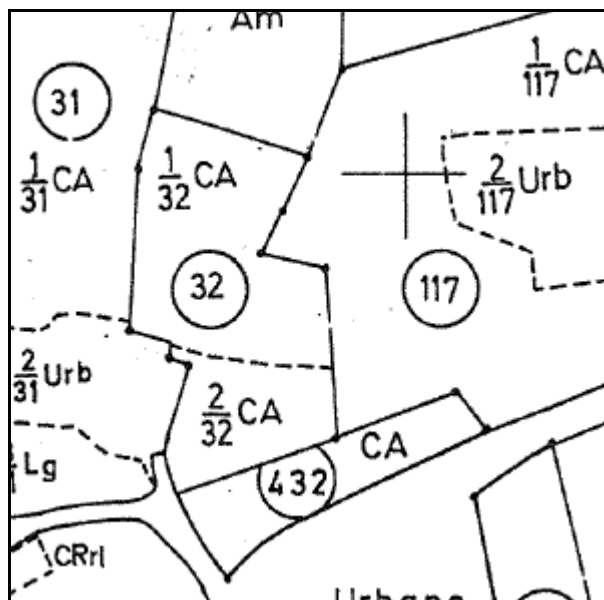


Figure 1 - Sample of a cadastral map (690x690, 300 dpi).

In the past years many administrative entities have decided to transfer the cadastral information to a numeric format and started using electronic management systems.

Only around 15% of the cost of a digital CIS (Cadastral Information System) is used in the hardware, 5% to the

software and 80% is consumed by the manual data acquisition from the paper maps.

Beside, the acquisition phase is not only expensive, but also time consuming (8 hours for an urban cadastral sheet) and the quantity of data to be transferred is huge (more than 100 000 sheets in Portugal – being one of the smallest countries in Europe).

Some work has already done analysing cadastral maps in other countries [1], but due to the formal aspects of representation, different approaches must be considered.

As a strategy the authors have opted for a heavy use of basic and well-known algorithms [2] with some eventual modifications. The first results are encouraging and about 75% of the information can be already extracted from the map.

2. PROCESS ORGANIZATION

Any Portuguese cadastral entity is composed by a closed contour, a numeric identification inserted in each parcel circle, possible existence of depended plots (small parcels), separation lines between parcels and finally the description of each parcel (Fig. 1). A single map contains many of these entities together with a meta-knowledge sector in which extra information about the map, usually modifications, are presented.

The authors have used all these information from the beginning to establish the global strategy of the analysis.

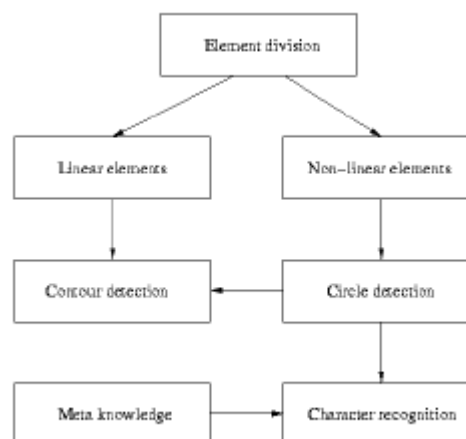


Figure 2 – Analysis process flow

For example, circles can provide an excellent departure point for analysis of an entity from inside. Figure 2 explains how the process can be guided by general knowledge of the domain.

The division between linear and non-linear elements is made analysing the occupied area. This knowledge permits the separated application of recognition algorithms. Recognized elements are marked to avoid interfering with another layer of analysis, and to resolve possible conflicts, making the overall process more reliable.

3. ANALYSIS EXAMPLES

Some important work was done for the analysis of the cadastral information. This includes circle, semi-circle, character recognition and contour detection. The following sections include some important remarks for each solution implemented.

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 have around 500 entities, 800 parcels, 2500 characters and a significant number of miscellaneous information).

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

A. CIRCLES AND SEMI-CIRCLES ANALYSIS

Some problems are associated with circle detection in a common cadastral map. The existence of circles, semi-circles (see Fig. 3), multi scaling (Fig. 4) and connection with elements defined as linear (see section II), makes the implementation of a robust circle and semi-circle detection algorithm more difficult.

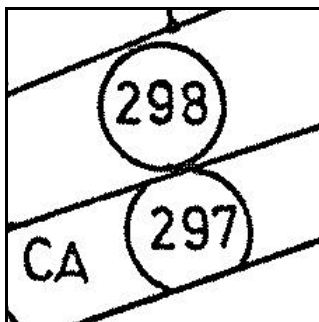


Figure 3 - Semi-circle, connection with linear object

Hough introduced some important work, with the known Hough transform [3]. Ballard [4] also presented a solid work based on Hough transform, which is a known

process for extraction of parametric shapes. The recognition is done by searching for global patterns in the image space by recognition of local patterns (point for example) in a transformed parameter space.

This work is very useful when the target patterns are sparsely digitised principally due to the existence of discontinuities or noise. The cadastral map is often of poor quality because its storage conditions and age.

The basic idea of this technique is to find curves that can be parameterised like straight lines, polynomials, circles, etc... in a suitable parameter space. Although the transform can be used in higher dimensions the main use is done in two dimensions to find, e.g. straight lines, centres of circles with a fixed radius. In our case, the existence of different scale circles makes the use of another dimension necessary. This makes the required parametric space very large, therefore the use of the standard Hough transform is unwise.

The parametric equations for a circle in polar coordinates can be seen in (1).

$$\begin{cases} x = a + r \cdot \cos(\mathbf{q}) \\ y = b + r \cdot \sin(\mathbf{q}) \end{cases} \quad (1)$$

Solving for the parameters of the circle we can easily obtain the equations, that relation the three dimensions (a, b and r) of the standard HT.

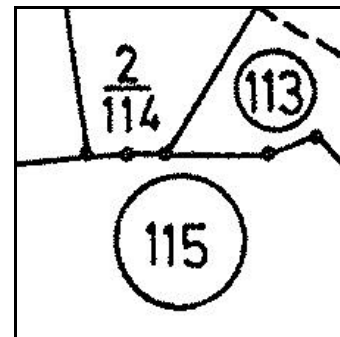


Figure 4 - Multi scaling

We use a simple, but efficient method based on a modified progressive probabilistic HT [5], for circle detection that also covers the recognition of semi-circles. To prevent the necessity of large storing and searching, local decision is made for each space point, resulting in strong circle mismatch recognition near a "true" circle. This problem is however easily resolved with a post-treatment searching and confirmation algorithm based on the known characteristics of the map and of the circles (such as: no overlay circles, possible radius values...).

The recognition is done progressively, making the algorithm fast and robust, using validation tools when some expected characteristic is under analysis. Next we explain the process recognition flow.

- From the division of elements presented in section II, the detection of complete circular elements over the non-linear class is very reliable, requiring only a few points of analysis, making the method very fast (high performance with accurate results were obtain for 15 points for a standard 300 dpi image).
- The recognition of semi-circular objects over the linear elements (connected semi-circles and possible complete circles), more points were used (35 points for 300 dpi).
- In case of a possible semi-circle, a validation algorithm is used, checking it's interior and boundary.

Table I presents some results for this methodology, obtained in a real map, with post-treatment analysis, that we consider very encouraging.

Shape	n° in image	Correct hits	Incorrect hits
Circle	294	89 %	0.013 %
Semi-Circle	84	81 %	0.038 %

Table I. Results obtained analysing a real map with 300 dpi

B. OPTICAL CHARACTER RECOGNITION

Optical character recognition (OCR) is one of the most widely used application of automatic pattern recognition and it's a very active research field since the 50's, in 1967 IBM started selling to big companies systems for OCR. Today we are able to recognize high quality text documents or especially written hand printed text, one practical example is the on-line recognition used by recent personal digital assistant like Palm or Compaq iPAQ.

Efforts are being made to achieve better recognition rate of degraded printed text and unconstrained handwritten text. The recognition speed becomes slower in these cases. In our application due to the amount of data to be processed, the recognition shouldn't be too time consuming.

OCR has, independent of the source of the characters, some steps that are common to most investigation projects like pre-processing the text to separate touching characters, a method to extract features from the characters, the classification, and in the end a post-processing for the validation of the resulting recognition, using a heuristic database of rules that takes in account the context where a given character was located.

Additionally optical character recognition has problems with characters that are incomplete due to noise or occlusions.

The segmentation problem of touching chars were solved by using projection histograms that try to detect minima in the middle from the horizontal histogram and maximal from the vertical histogram. This is done because most of the touching chars are the parcels fraction number that could be observed in Fig. 5.

This works rather well. After the first split the separated characters are tested again and if the resultant characters aren't recognized another split is tried with a limit of possible split alternatives that are most frequent.

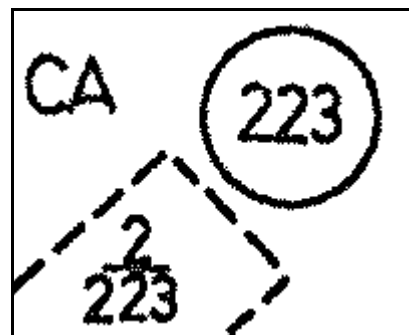


Fig. 5 - Connected character's problems

After analysing different feature extraction methods [6] like: template matching, deformable templates, unitary image transforms, graph description, projection histograms, zoning, geometric moment invariants, spline curve approximation, Fourier descriptors, we chosen the template matching due to its simplicity, taken in account that the speed is a important factor but also because we have a pre-defined font and properties like rotation or scale invariance is easily achieved without complex computation required with some of previous mentioned methods.

Instead of using all the template points, only important points were selected. Later weights (+, -) were attributed to these points acting as confirmation or rejection points [7]. This information was stored in a list with the relative point positions and the associated weights; Fig. 6 shows two examples for the list point's position. From these four examples, it can be seen that the weights should be carefully chosen, because the most significant differences should be highlighted.

For the positive weights a skeletonization of the character was done, and equidistant points were chosen. For the negative weights two different dilations were done, and the difference between the bigger and the

smaller were stored. Similarly we selected equidistant points and stored them in a list with the respective negative weights.

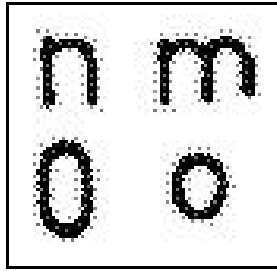


Fig. 6 – Example of similar character distinction

The initial method used to attribute the weights was a heuristic knowledge of representative points from each character. To solve this cumbersome manual method a genetic algorithm was introduced that gives the optimal weight list for each character. This algorithm uses a fitness function that counts the number of true or false recognitions in a database with multiple examples of every possible character.

Although a list is being used, we are planning to use a neural network to compare the recognition rate and speed with our current implementation. The chosen network was a Multiple Layer Perceptron with Back Propagation were two previous steps are required. First the segmentation that is already implemented for the weight list and a second step, a normalization of characters is necessary due to the fixed number of inputs from the neural network. This is done using splines with bicubic interpolation (Eq.2).

$$F(p',q') = \sum_{m=-1}^2 \sum_{n=-1}^2 F(p+m,q+n) R_c(m-a)R_c(-(n-b))$$

Equation 2 - Bicubic Interpolation

In Equation 2 function $F(x, y)$ is the image pixels and $R(x)$ is the interpolation function that in this case is a spline.

C. CONTOUR DETECTION

Contour detection is an important task in automatic recognition of cadastral maps, mainly because it gives the coordinates of points that constitute a parcel into a GIS.

These points may be two coordinates of a line segment, or three points of a Bezier curve.

The main problem in contour detection is the existence of discontinuities. These discontinuities provoked by bad

scanning or by the effect of noise. Two strategies are possible to approach this problem. To restore discontinuities or to use algorithms that aren't sensible to it. To restore discontinuities, may be used line following algorithms. The problem of this kind of algorithms is when there are interceptions of parcels, which line to follow. This is a decision problem, with many implications problems, so the authors decided to drop this option.

One can use algorithms not sensible to discontinuities. Initially it was intended to do this analysis using active contour model, also known as Snakes [8], but due to a lack of different energies fields, it wasn't appropriated to apply in this case. This detection is based on the knowledge provided by the circle position already detected, as refereed in section I.

As every parcel contains only one circle inside, this information is reliable.

Before applying the contour detection, the map should be pre-processed, this means that non-linear elements should be eliminated. This is important for the application of our algorithm of automatic contour detection, as it will be seen.

```

DetectContours(Image,Circles,N){
  for i=1:size(Circles),
    CritPoints=Fill(Image,Circles(i).x,Circles(i).y,N)
    Contours(i)=BrushAlgorithm(Image,N,CritPoints)
  }
  return(Contours)
}

Fill(Image,Circle,N){
  if(EmptySquare(Image,x,y,N))
    FillSquare(Image,x,y,N,GrayColorSquare)
    Fill(Image,x-N,y,N)
    Fill(Image,x+N,y,N)
    Fill(Image,x,y-N,N)
    Fill(Image,x,y+N,N)
  else
    CriticalPoints.add(x,y)
    return(CriticalPoints)
  }

BrushAlgorithm(Image,N,CritPoints){
  if(GraySquareLeft)
    for y=1:N,
      for x=1:2N, // Scroll
        if(!LockX)
          if(Black)
            Contour.add(x,y)
            Paint(x,y,LockColor)
            LockX=True
          else
            if(LockColor)
              LockX=True
            LockX=False // Restart Lock locally
            LockX=False // Restart Lock globally
            // Continue to bottom,right and top ...
            return(Contour)

```

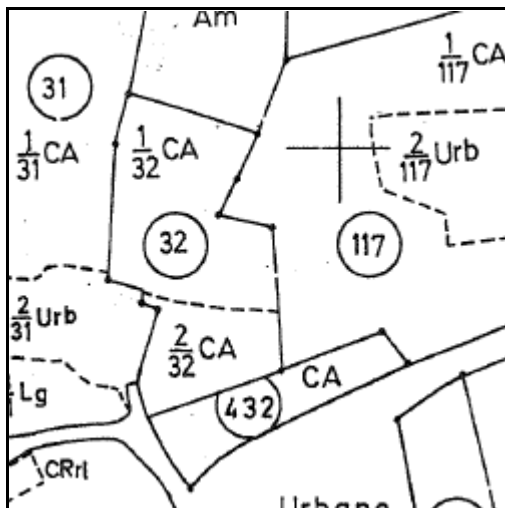
Algorithm 1 – Automatic contour detection

Creation of an innovative algorithm (see algorithm 1), with no sensibility to discontinuities and with the initial knowledge of a point inside the contour was necessary for our specific problem. Fill algorithm is a valid process but it's sensible to discontinuities of contour, so it should be reformulated.

Instead of using a normal fill, a fill with a variable block size was used. In this way, using the recursivity, blocks that are part of contour are identified and then processed locally, these areas are called *critical squares*.

detection of edges should be made from all four directions.

Figure 7, is a representative example of the presented algorithm, in (a) the original image is presented. After processed, figure (b) was obtained and in figure (c) there can be seen the application of Brush algorithm. The result of contour detection is represented in (d). To perform a automatic contour detection, the size of an generic block (N) must be determined.



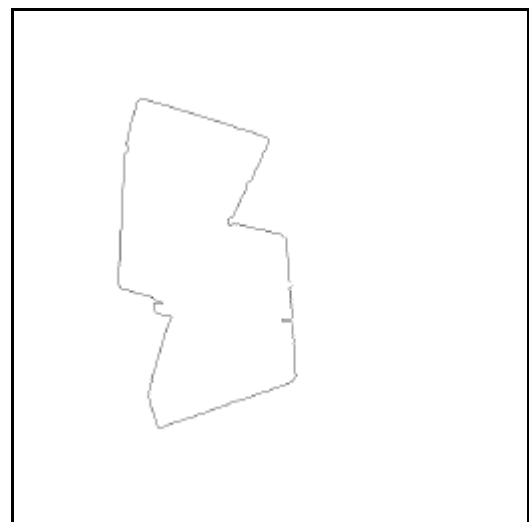
(a) Original Image



(b) Processed Image



(c) Brush algorithm



(d) Final result

To locally process these critical zones it's considered the squares neighbourhood was considered. To do this a list of critical squares is made and later, all filled neighbours are processed.

When the neighbours have filled squares, the process of contour detection starts. This method is called *Brush* method because it's like a "paint" brush. After detection with a contour point, a match happens, and its axis is locked. Effectively contour edges are obtained, so the

This can be done applying a fill in each circle position and then a search is made for the smallest block size that satisfy the non existence of any empty contour. This happens because if it's made a fill in one parcel, that goes outside it's contour, then other parcel gets a empty contour.

4. FUTURE WORK

Higher-level analysis will be integrated into the system to be responsible for the semantics of a map. This analysis will create relations between different objects. For example, between an urban pattern (a building) and its land plot or between a series of digits and the number they represent.

They will look for conflicts between lower-level results. For example, whether a building is marked with two plot numbers. They are also involved when information is missing. For instance if a land plot is recognized, it should have a plot-number. If this is not verified, as the number is usually put close to the plot, search agents will first search for the number fields in the unrecognized area list and if they still don't find the number, they will request another refined analysis of this region. If the expected information remains unfound, then a question will be asked to the user. Hence, the system verifies possible conflicts or collisions only in a first iteration, which means that the solution is established by only changing the status of the directly neighboring objects.

This high level process needs a control schema such as a Blackboard for example. This architecture will allow the system to use many different and heterogeneous methods together to achieve better performances.

5. CONCLUSION

A conventional vision system is composed of four major tasks: (1) pre-processing (noise reduction and edge enhancing), (2) segmentation, (3) object recognition and (4) scene understanding. An alternative approach has been proposed in this paper. Robustness to noise and occlusions makes pre-processing useless. It was shown that real images can be analyzed directly and this saves much CPU time. As no global segmentation is needed, the system does not lose information.

Object recognition and scene understanding can be done together. When a human observer sees an object and cannot say what it is, he/she uses other objects or colours surrounding this object and takes a decision according to his/her experience.

What is important in a BB architecture is that all agents are independent and an agent can easily be added or removed, and different analysis methods can be used together. Each new requirement can be easily integrated into the system.

The low-level but robust recognition methods and the BB architecture used together appear to give very good results when working with hand-drawn documents. Many time-consuming processes can be avoided by the use of all contextual information available.

6. 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.

7. REFERENCES

- [1] H. Shahbazkia, *Reconnaissance invariante et acquisition de connaissance: application au traitement automatique des plans de cadastre Français*, PhD thesis, Universit Louis Pasteur de Strasbourg, 1998.
- [2] K. Tomber, C. Ah-Soon, Ph. Dosch, A. Habed and G. Masini, "Stable, Robust and Off-the-Shelf Methods for Graphics Recognition", *In Proceedings of the 14th International Conference on Pattern Recognition*, Brisbane (Australia), pp. 406--408, 1998.
- [3] P. V. C. Hough, "Method and means for recognizing complex patterns", *US Patents 3069654*, 1962.
- [4] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes", vol.~13, no.~2, pp.~111--122, 1981.
- [5] C. Galambos J. Matas and J. Kittler, "Progressive probabilistic Hough transform", Technical Report, University of Surrey/Czech Technical University, 1998.
- [6] O. Trier, A. Jain, and T. Taxt, "Feature extraction methods for character recognition - a survey", 1996.
- [7] P. Gader, B. Forester, M. Ganzberger, A. Gillies, B. Mitchell, M. Whalen, and T. Yocum, "Recognition of handwritten digits using template and model matching", *Pattern Recognition*, vol. 24, pp. 421--431, 1991.
- [8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models", *Proc. of IEEE Conference on Computer Vision*, London, England, 8-11 1987, pp. 259-268.