

Geometric Proportional Analogies In Topographic Maps: Theory and Application

Emma-Claire Mullally, Diarmuid P. O'Donoghue,
Department of Computer Science / National Centre for Geocomputation
National University of Ireland, Maynooth, Ireland.
{emmaclaire.mullally;diarmuid.odonoghue}@nuim.ie
www.cs.nuim.ie/~dod / www.nuim.ie/ncg

Amy J. Bohan, Mark T. Keane,
Department of Computer Science
University College Dublin, Ireland
{amy.bohan;mark.keane}@ucd.ie
www.cs.ucd.ie

Abstract

This paper details the application of geometric proportional analogies in the sub-classification of polygons within a topographic (land cover) map. The first part of this paper concerns geometric proportional analogies that include attributes (fill-pattern and fill-colour). We describe an extension to the standard theory of analogy that incorporates attributes into the analogical mapping process. We identify two variants on this attribute matching extension, which is required to solve different types of geometric proportional analogy problems. In the second part of this paper we describe how we adapt one of the algorithms to identify structures in topographic datasets (maps). We detail the results of identifying a number of different structures on a sample topographic map.

1. Introduction

Sub-classification of polygon objects in maps can be used to improve the quality of map data. For example *roads* can be sub-categorised as *cul-de-sacs* or *Y-junctions* and *buildings* can be sub-categorised as *houses* or *schools* etc. Currently, OS MasterMap¹ data can be thought of as thousands of polygons with only a minimum of information supplied about each polygon. Firstly a topographic polygon is an area bounded by a set of lines. Secondly each polygon is categorised as one of thirteen main themes including *road-or-track*, *rail*, *river*, *roadside*, *general surface*, *building* etc. Map users are expected to decipher exactly what real world object each polygon corresponds to, for example is a road a *crossroads (X-junction)*, a *cul-de-sac* or a *T-junction*? In order for these maps to be truly useful,

¹ Ordnance Survey Research Centre, Southampton, UK.

as much information as possible must be given to each polygon. It is more useful to know that a polygon is a *hospital* than it is to know only that it is a *building* of some description. A map is of little use if we cannot successfully identify what is on it. This is a particular problem for deriving automated solutions, such as route planning. It is difficult, if not impossible, to automatically detect something on a map when nothing on the map is clearly identified in the first place. Another problem to be considered is that this information should be easily available not only to people, but also to computational processes. For example, the automatic identification of *houses* in a map that are located in a *cul-de-sac*, could assist house buyers that are searching for such a property. Also automatic identification of *Y-junctions* (or *T-junctions*) and *X-junctions* may assist in deciding where to put a set of traffic lights for town planners.

The context of a polygon refers to the other polygons (neighbours) that share a boundary line or boundary point with that central polygon. This context can prove invaluable in sub-categorising polygons. We describe a method of identifying subcategories of polygons using their context. This is achieved by *analogical matching* [1, 2, 3, 4] to a previously defined polygon context. For example, a road network is composed of many individual *road* polygons or segments, which are adjacent to one another, forming a network. Some of these polygons may in fact be junctions. If a *road* polygon is surrounded on three sides by *roadside* polygons and is adjacent to only one other *road* polygon, we may infer that this central polygon is actually a *cul-de-sac*.

In this paper we detail the application of analogies in sub-classifying *roads* as *cul-de-sacs*, *Y/T-junctions* and *X-junctions*. To test our application we use a segment of the topographic layer of the OS MasterMap, covering some of the region of Port Talbot. Before describing the application it is necessary to give an overview of geometric proportional analogies, which are the analogical comparisons used in the classification process. Section two therefore details the theoretical aspect of analogy and Section three describes how this theory can be put into practice when sub-categorising roads.

2. Analogical Comparisons

In this section we examine geometric proportional analogies (GPAs) and discuss how they may be solved algorithmically. GPAs include those analogy problems commonly found in IQ tests. We then present a different type of GPA problem that includes attributes within the analogy process. We identify two distinct algorithms that are required to solve two different classes of GPA problems. In the following section we then develop one of these algorithms and illustrate its application to processing topographic maps. However, we begin by looking at plain GPAs.

2.1 Geometric Proportional Analogy (GPA)

Analogies are comparisons between two systems of information, referred to as the *source* and *target* domains. GPAs are a type of analogy formed between two collections of geometric figures. They are of the form $A:B::C:D$, which we read as

A is-to B as C is-to D. The *source* domain (A:B) identifies some transformation(s), which must then be applied to C, yielding D (See Figure 1). For example, the analogy in Figure 1 centers on partitioning the central polygon of part A to produce part B. This partitioning transformation must then be applied to C and this allows us to generate D.

There are two key points to note about GPAs. First, the change in the source domain (*i.e.* between A and B) is called the *transformation*. Second, parts A and C are used to identify the inter-domain *mapping* [1] *i.e.* small square (A) maps to small circle (C) and central square (A) maps to central circle (C). The combination of the transformation and mapping will yield the solution D.

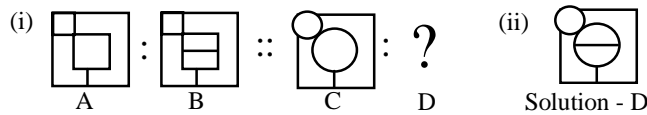


Figure 1: A Simple Geometric Proportional Analogy

2.2 Theoretical Background

Evan's ANALOGY [5] model was the first computational model to solve GPA problems. It focused on geometric analogies involving plain figures, taking graphic images as input and making use of the geometry of the presented figures. Additionally, Evan's model selected the best D from a given list of alternative solutions.

More recent models of GPAs have made use of Gentner's structure mapping theory [1]. This states that all analogies may be solved by representing the source and target domains using predicates. Then the crucial inter-domain mapping between the source and target domains can be discovered by identifying the largest isomorphic mapping between the predicate structure of the two domain descriptions. Structure mapping theory has been successfully used in many models of conceptual analogies [6, 7, 8] as well as GPAs [6, 9, 10].

Previous models of GPA's determine the solution by selecting the best of a number of presented answers such as Tomai & Evans [5, 6]. The key distinction of our model is that it actually generates a description of the desired solution. This generation step will prove a key point when the geometric analogy algorithm is migrated to solving problems within topographic maps. This solution generation step will be elaborated later in the paper. Also in solving these GPA's we concentrate on the topology of parts A, B and C rather than the geometry of the figures they contain. That is, we ignore the shape of the individual polygons and concentrate on their topological arrangement. The following section details our model for solving GPAs.

2.3 Analogical Mapping

Our solution to GPAs [9] uses a predicate description of parts A, B and C of the analogy. Our representation takes regions as primitives, with points and lines only being admitted in so far as they define the bounds of these regions. We uniquely identify each such region in both the source and target domains (see Figure 2).

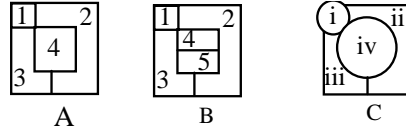


Figure 2: Labeling objects in the source and target domains

Having identified and labeled the distinct objects in the domain, we characterize the structure within each part (A, B, C) using two binary relations: *line-adjacent*(x,y) and *point-adjacent*(y,z).

Line adjacent polygons are polygons that share at least one edge with the current polygon. Point adjacent polygons are polygons that are not line adjacent to the current polygon but that touch it at one or more points.



Figure 3 Line Adjacent Neighbours



Figure 4 Point Adjacent Neighbours

Part A of Figure 1 is therefore represented by the following collection of predicates, based on the labeling shown in Figure 2.

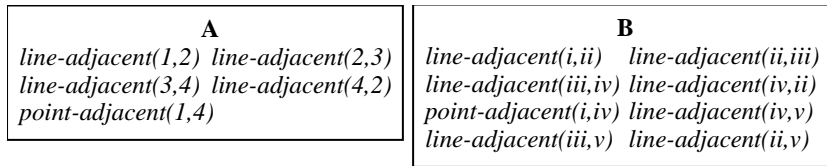


Figure 5: Representation of the source domain of Figure 1

Next, we perform the central structure mapping step [1] that identifies the inter-domain *mapping* between the descriptions of part A and part C. (As the counterpart of B has yet to be generated, we ignore parts B and D at this stage). The inter-domain mapping consists of pairs of objects and relations, and these paired items represent counterparts of one another. So, object 1 in A is the counterpart of object i in C. This structure mappings process concentrates on the topological relations between objects, allowing different shaped polygons be placed in correspondence between parts A and C. So, the *square* 1 of part A is mapped to *circle* i of part C (see Figure 1).

2.3.1 The Source Domain Transformation

The most important aspect of the source domain is the *transformation* it defines between parts A and B. This transformation is represented implicitly by the difference between the descriptions of these two parts. (Unlike [10] we do not use a separate vocabulary to represent the source domain transformation). The transformation in Figure 1 involves the insertion of an extra polygon by splitting an existing polygon. Therefore the solution to this problem is the application of this “insertion” transformation to the mapped equivalent in C – yielding D in Figure 1(ii).

Other GPA problems contain transformations in the source domain that insert or delete polygons from the source (and thus the target), or perform other modifications to it. GPA’s may also perform translations and rotations on the objects or may replace an object by a different shape. However, our application domain of topographic maps makes best use of analogies that are translation, rotation, scale and shape invariant. That is, we wish to reason about topographic structures in different places and with objects of different shapes. Therefore, in this paper we focus on GPA’s that are most similar to those we used in processing topographic maps.

We employ the standard algorithm for generating analogical inferences, which performs pattern completion on the information supplied in parts A, B and C to generate D. This is also known as the “Copy With Substitution and Generation” (CWSG) algorithm [11]. So, to generate the description of D we copy the structure of part B – but we substitute the object names of B with their mapped equivalents. Item 5 has no counterpart in the mapping, and so an appropriate token is generated for this in the description of D (such as *v*).

The source defines no transformation to the predicate description and the source (A) and target (C) predicate information is identical. We can copy the predicates in part C to generate the predicate description for the required solution D. However, the object identifiers must be replaced by their mapped equivalents from C.

However, the application domain of topographic maps is not composed of undifferentiated polygons, but rather is composed of polygons each with a specific category. Incorporating this additional category information can be achieved by treating each category as an attribute of the relevant object. With this objective in mind, we explore geometric analogies that contain additional information.

2.4 Attributes in Geometric Proportional Analogies

We now describe a different type of GPA problem, which has not been addressed by previous models. These are GPA’s that involve attribute transformations and attribute matching (see Figure 6). The attributes contained in these problems are vital to their solution and thus the attribute information cannot be overlooked. Keane *et al* [8] explored analogies involving attributes, but this work did not focus on proportional geometric analogies nor did it address attribute transformations. None of the existing models of geometric analogies looks to solve GPA problems such as that depicted in Figure 6.

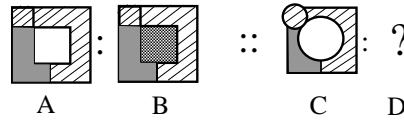


Figure 6: A Proportional Geometric Analogy involving Attributes

For example in Part A of Figure 6, we can see that one object is plain, two have a texture of diagonal lines and another object is grey. Because qualities like *colour* and *texture* are associated with individual objects, this information is represented by single place predicates – or attributes. So, we add the attribute information about each polygon to the predicate descriptions (from above).

striped(1), striped(2), grey(3), plain(4).

Of course, we still identify the inter-domain mapping between the predicates of the source and target domains – or between part A and part C of Figure 6. This mapping process ignores the attribute information, and the attribute information is only utilised after the mapping has been identified. The presence of attribute information does necessitate an extra degree step in the solution process.

2.4.1 Attribute Transformations & Attribute Matching

The source domain (A:B) of Figure 6 includes a transformation to its attribute information, specifically to the *fill-pattern* attributes. As with the predicate information, the attribute transformation is represented implicitly by the difference in attributes in parts A and B. The attribute transformations (implicitly) defined by the source domain of this problem is:

A: *striped(1), striped(2), grey(3), plain(4)*

B: *striped(1), striped(2), grey(3), chequered(4)*

The attribute information from part C is added to its earlier predicate description. This information will prove crucial to the next attribute matching step, required to generate the correct D. The attributes added to predicate description of part C are:

C: *striped(i), striped(ii), grey(iii), plain(iv)*

2.4.2 Global Attribute Matches

We define *Attribute Matching* as the process of determining the attribute changes that occur across the transformation and mapping process. *Attribute matches* occur between the attributes of objects that map with one another. Therefore, attribute matching must occur after the structure mapping process has taken place. So the attributes of object 1 in the source, will be matched with the attributes of object i in the target, and so on for the other objects.

Attribute matching can manifest itself in two main ways: *Global attribute matches* and *Local attribute matches*. This distinction arises because there are multiple ways of identifying the mapping between these attribute transformations. We begin with the simpler form of attribute matching - global attribute matching. We define

a *global attribute match* as a match between attributes where there is a 1-to-1 correspondence between all attributes in the source and target domains.

Now we identify the 1-to-1 correspondence between the attributes of parts A and C. So, *striped(1)* matches with *striped(i)* etc. Table 1 details the attribute matching required for the problem in Figure 6. In this problem, the attributes that are mapped together are in fact, also identical. These are among the easier class of problem to involve attributes. Crucially, this is also the type of attribute matching that we shall make use of in our application for processing topographic maps.

Table 1 Global Attribute Matching

Global Attribute Transformation			
Source		Target	
A	B	C	D
<i>striped(1)</i>	-> <i>striped(1)</i>	<i>striped(i)</i>	->
<i>striped(2)</i>	-> <i>striped(2)</i>	<i>striped(ii)</i>	->
<i>grey(3)</i>	-> <i>grey(3)</i>	<i>grey(iii)</i>	->
<i>plain(4)</i>	-> <i>checkered(4)</i>	<i>plain(iv)</i>	->

Figure 6 may be solved by use of the global attribute matching, generating the solution depicted in Figure 7.

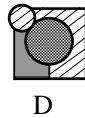


Figure 7: Solution to the Geometric Proportional Analogy in Figure 6

There is also a local *attribute matching algorithm* that is required to solve more complex GPA problems. This algorithm is required in cases where there are multiple matchings required for an attribute across different objects. However, we have found that most problems in topographic maps can also be solved using global attribute matches. However, the possibility of more complex local attribute matches occurring can not be ruled out. For further information on local attribute matching see [9].

3. Geometric Proportional Analogies in Topographic Maps

We now apply our geometric proportional analogies to the task of processing topographic maps. A topographic map can be thought of as a large collection of (mostly) non-overlapping polygons describing homogenous collections of land cover across an area. OS UK categorise polygons into one of approximately 13 themes (categories), including: *road-or-track*, *rail*, *river*, *roadside*, *building* and *general surface*. However, many of these categories are very broad. The remainder of this paper centres on (automatically) improving the quality of the data recorded on OS UK's MasterMap map of Great Britain.

Figure 7 depicts a source domain representing a T-shaped collection of polygons, which are all from the same category. The attributes of the central polygon are modified by the source domain transformation of the GPA. The target domain (C) represents a collection of polygons located within a topographic map. Three of these polygons are from the *road* category (theme).

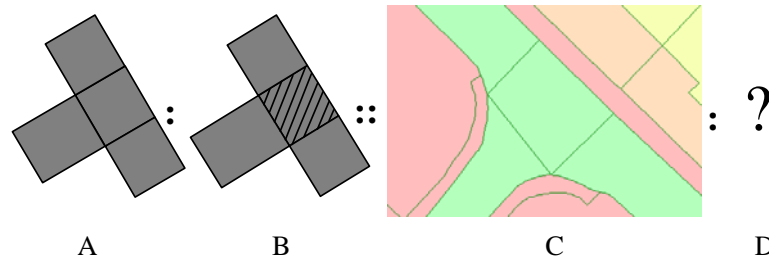


Figure 8 A Geometric Proportional Analogy within a Topographic Map

3.1 How do we find the cul-de-sacs?

A *cul-de-sac*, or dead end, is a road that is blocked off and can only be accessed through one entrance. They are often found in housing estates, but they can in fact be found anywhere, in industrial estates, hospitals, in the countryside and the city centre.

It is useful to know where cul-de-sacs are for several reasons. It is a commonly held belief that houses situated in a *cul-de-sac* tend to be more expensive than houses that are situated on the main *road*. This is largely due to the fact that *cul-de-sacs* are seen as being quieter, with less traffic and are safer for pedestrians and children. Cul-de-sacs may be useful in route planning

A cul-de-sac can be thought of as a *road* segment (polygon) that is connected to only one other *road* segment. In this way the task can be reduced to finding all of the “*road or track*” polygons in a map that are line adjacent to only one other “*road or track*” polygon and identifying each as a *cul-de-sac*. The shape or size of the polygons does not need to be taken into account. If a road segment is *line-adjacent* to two or more different road segments, it is not considered a cul-de-sac.

In order to find a *cul-de-sac*, first we generate a list of *line-adjacent* polygons for each individual polygon in the map. In order to be certain that we have found a cul-de-sac, the list of polygons that are *line-adjacent* to the “road or track” polygon in question must be checked and the numbers of “road or track” polygons that appear in its *line-adjacent* list are counted. If this total is equal to one, then we have found a cul-de-sac. This task does not require us to consider *point-adjacent* polygons.

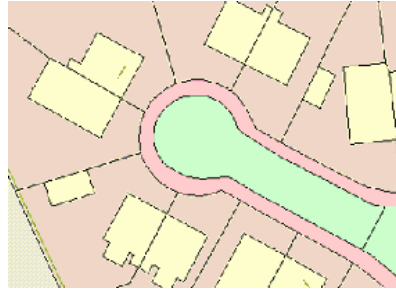


Figure 9. A cul-de-sac within a housing estate

Comment: Do we actually do this? If not, then maybe we shouldn't be talking about it – unless it

Sometimes, in looking for cul-de-sacs, errors occur due to rivers or railways crossing over a road. This is known as the occlusion problem. These occlusions can be removed before categorising the cul-de-sacs by finding where two roads are separated by only a river or railway polygon and replacing this occluding polygon with another road polygon [12].

It is also interesting to note that because duplicate line and point adjacencies are removed by the CSM algorithm, we do not have any problem with categorising cul-de-sacs that are adjoined to the same road twice, an example of which can be seen in Figure 3.

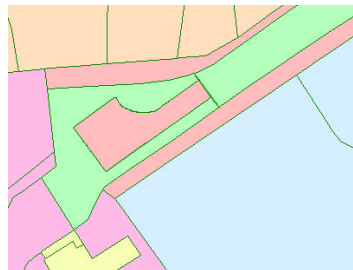


Figure 10 A cul-de-sac line-adjacent to the same road twice

3.1.1 Finding Y-junctions and X-junctions

In a similar manner to how we found cul-de-sacs we can also find Y-junctions and X-junctions.

A Y-junction, or T-junction, is a road segment where there is a choice of two different routes that can be taken. It can be thought of as a road segment that is line adjacent to three other road segments. In order to find all of the Y-junctions in a map, we use the same algorithm that we used to find the *cul-de-sacs* and we check if the total number of road-or-track polygons that are *line-adjacent* to a road-or-track polygon being considered is equal to three. If this is the case we have found a Y-junction.

An X-junction is a road segment where there is a choice of three different routes that can be taken. We define it as a road segment that is line-adjacent to four other

road segments. We find X-junctions in the same manner as we find Y-junctions, except that we look for the polygons that are line-adjacent to a total of four other road-or-track polygons.



Figure 12 X-junction

If a road-or-track segment is found that is line-adjacent to two road-or-track polygons, or if it is line-adjacent to more than four road-or-track polygons, it is discarded.

3.2 Tests on the “Port Talbot” Dataset

The results below are based on the OS MasterMap data for Port Talbot, which is mainly a residential area containing over 5000 non-overlapping independent polygons. For our purposes we have looked only at the topography layer, which contains large-scale data of the town’s real world surface features, to find cul-de-sacs, Y-junctions and X-junctions. This is a convenient data form to use as it contains a lot of detail about features in the area and their adjacencies. Examining the data is also quite simple and road segments can be found with relative ease.

We have decided to omit polygons that are on the edge of the map from our testing, as these polygons would produce unreliable results. We have also omitted possible occlusions. It is possible to remove occlusions before sub-categorising cul-de-sacs.

In each case, all of the cul-de-sacs, Y-junctions and X-junctions in the map were verified independently and systematically by hand, using two human map readers. First, each *cul-de-sac*, *Y-junction* and *X-junction* in the map was found manually, and separately, by both map readers. It was confirmed that they had both found the same number of objects in each category. Secondly, we used the algorithm detailed in section 3.1 to find each instance of a cul-de-sac, Y-junction and X-junction in the map. Finally, we compared the map readers results against the results derived from using our algorithm. This was carried out to assess the accuracy of our algorithm at identifying all members of each category. We also wanted to ensure that our definition of each category did not deviate from what the map readers perceived to be a cul-de-sac, Y-junction or X-junction.

In order to analyse our results we will look at them in terms of *precision*, *recall* and *F-measure* [13]. In this case, the *precision* is the ratio of the number of correctly identified road segments to the total number of road segments returned by our algorithm.

$$precision = \frac{\text{number of correct answers given}}{\text{total number of answers given}}$$

Recall is the ratio of the total number of correct answers returned from a dataset, over the total number of correct answers contained in the dataset.

$$recall = \frac{\text{number of correct answers given}}{\text{total number of possible correct answers}}$$

The *F-measure* combines the values of both precision and recall. The beta (β) is a parameter used to assign the relative importance of precision to recall. In our example, we have allowed beta to assume a value of 1, meaning that precision and recall are given equal importance.

$$F - \text{measure} = \frac{(\beta^2 + 1) \text{precision} * \text{recall}}{\beta^2 \text{precision} + \text{recall}}$$

Table 2 Accuracy of the Sub-classification tasks

	Quantity	Precision	Recall	F-measure
Cul-De-Sac	36	100%	100%	1
Y-junction	87	93.5%	100%	0.966
X-junction	10	91%	100%	0.953

3.2.1 Cul-de-sac Results

Our results indicate that, by our definition, there are 36 cul-de-sacs in Port Talbot. Further inspection by hand concluded that all of the polygons returned do, in fact, correspond to cul-de-sacs.

Looking at these results in terms of precision, the total number of correctly identified cul-de-sacs was 36 and the total number of answers returned was also 36. This gives us a precision of 100%, which means that every cul-de-sac that was returned was correctly identified as a cul-de-sac. There was no case in which a polygon was incorrectly identified as a cul-de-sac.

With regard to recall, the number of correct answers returned for our definition of a cul-de-sac was 36. The total number of cul-de-sacs in Port Talbot is also 36. This gives us a recall of 100%, which means that every cul-de-sac in Port Talbot was identified.

The F-measure for cul-de-sacs is equal to one. This is to be expected, as both precision and recall are also equal to 100%.

3.2.2 Y-junctions Results

Y-junctions are defined as a road segment that is line adjacent to three other road segments. The number of Y-junctions found manually is 87 and the total number of answers returned by our algorithm is 93. This gives us a precision of 93.5%.

The incorrect polygons are slip-roads onto a motorway. In the future we hope to be able to identify the motorway and exclude it from our testing in order to improve results. The results returned by our algorithm include every Y-junction that exists in the Port Talbot dataset.

The total number of correct answers given by our algorithm, 87, was equal to the total number of possible correct answers, and so the recall is 100%. We can be sure that no Y-junction exists in the map that does not appear in our list of results. Given the precision and recall, the F-measure for Y-junctions is 0.966.

3.2.3 X-junctions Results

The definition of an X-junction is a road segment that is line adjacent to four other road segments. Although all of the eleven road segments that are returned as part of our results are adjacent to four other road segments, one of the results could not be classed as an X-junction by our map readers as it is a motorway entrance.

Looking at the number of correct answers found manually, 10, against the total number of answers returned by our algorithm, 11, gives us a precision of 91%.

In total there are 10 X-junctions in the Port Talbot map. All ten of these junctions were found, and so we have 100% recall.

The F-measure for X-junctions, given the precision and recall, is 0.953.

4. Conclusion

In the beginning of this paper we set out to show that road segments in topographic maps could be successfully sub-categorised from the current *road-or-track* category into the sub-categories of *cul-de-sac*, *Y-junction* and *X-junction*. This can be found by looking at the geometric analogies, along with attributes, in topographic maps. The results have shown that that this method works extremely well for sub-categorising cul-de-sacs as well as different types of junctions, such as Y-junctions and X-junctions. By ignoring the shape of the polygons and concentrating on the local topology between polygons, i.e. the neighbouring polygons in the map, we have reduced this complex problem into something that is simpler to represent.

We examined proportional geometric analogies that include attribute information (fill-colour, fill-pattern etc) in the analogy process. We then described the attribute matching extension to the standard structure mapping algorithm employed by virtually all models of the analogy process. We specified two variants on this structure matching extension that can be use to solve different types of these geometric analogy problem. We then described how one of these algorithms was adapted to identifying localised collections of polygons within topographic maps.

Our contextual structure matching (CSM) algorithm appears to be a successful method for classifying different types of road segments, namely *cul-de-sacs*, *Y-junctions* and *X-junctions*. In fact, by using this method, cul-de-sacs, Y-junctions and X-junctions were sub-categorised with over 90% accuracy. In the case of cul-de-sacs the results could not be any better than 100%, although this figure may drop when a different topographic maps are used. Preliminary results (on other

maps) suggest that, for cul-de-sacs, the accuracy should not drop below 90%. In the case of Y-junctions and X-junctions, there was only one object on the map that caused incorrect data to be returned. However, using further sub-categorisation experiments, we may be able to fix this problem by identifying the motorway and excluding it from all other Y-junction and X-junction calculations. It is hoped that the more sub-categorisations of roads that we can find, the more accurate the results will become.

References

1. Gentner, D. "Structure-Mapping: A Theoretical Framework for Analogy", *Cognitive Science*, 7, 155-170, 1983.
2. Mulhare, L. O'Donoghue, D. Winstanley A. C. "Analogical Structure Matching on Cartographic Data", 12th Artificial Intelligence and Cognitive Science AICS-2001, NUI Maynooth, Ireland, pp 43-53, Sept. 5-7, 2001, ISBN 0-901519-48-0,.
3. O'Donoghue, D. Winstanley, A.C. "Finding Analogous Structures in Cartographic Data", 4th AGILE Conference on G.I.S. in Europe, Czech Republic, April, 2001,.
4. Winstanley, A.C. O'Donoghue, D. and Keyes L. "Topographical Object Recognition through Structural Mapping", 1st International Conference on Geographic Information Science - GIScience 2000, Savannah, Georgia, USA, October 28-31, 2000.
5. Evans, T.G "A Program for the Solution of a Class of Geometric Analogy Intelligence-Test Questions", in *Semantic Information Processing*, (Ed.) M. Minsky, MIT Press, 1967.
6. Tomai, E. Forbus Kenneth D. Usher, J. "Qualitative Spatial Reasoning for Geometric Analogies", 18th International Workshop on Qualitative Reasoning, August 2-4, 2004, Northwestern University, Evanston, Illinois, USA (during 26th Annual Meeting of the Cognitive Science Society).
7. Falkenhainer, B. Forbus, K.D. Gentner, D. "The Structure Mapping Engine: Algorithm and Examples", *Artificial Intelligence*, 41, 1-63, 1989.
8. Keane, M. T. Ledgeway T. and Duff, S. "Constraints on Analogical Mapping: A Comparison of Three Models", *Cognitive Science* 18, 387-438 (1994)
9. Bohan, A. O'Donoghue D. "A Model for Geometric Analogies using Attribute Matching", AICS-2000 11th Artificial Intelligence and Cognitive Science Conference, Aug. 23-25, NUI Galway, Ireland, 2000.
10. Davies, J. Goel, A. K. "Visual analogy in problem solving", *Proceedings of the International Joint Conference on Artificial Intelligence 2001*, Morgan Kaufmann publishers, pp 377-382, 2001.
11. Holyoak K. J. Novick L. Melz E. "Component Processes in Analogical Transfer: Mapping, Pattern Completion and Adaptation", in *Analogy, Metaphor and Reminding*, Eds. Barnden and Holyoak, Ablex, Norwood, NJ: 1994.
12. O'Donoghue, D. Loughlin, A. "Using Context to Repair Partial Occlusions in Topographic Data", *Proceedings of GISRUK - Geographical Information Science Research Conference*, Norwich, UK, pp 15-18, 28-30 April, 2004.
13. Jurafsky D. & Martin J, "Speech and Language Processing", Prentice Hall, 2000.