Identifying Attributes for Characterizing Game Area Types in Virtual Terrain

Andrew Pech*
Edith Cowan University
270 Joondalup Drive
Joondalup WA 6027 Australia
apech@our.ecu.edu.au

Philip Hingston, Martin Masek, Chiou-Peng Lam
Edith Cowan University
270 Joondalup Drive
Joondalup WA 6027 Australia
{p.hingston, m.masek, c.lam}@ecu.edu.au

ABSTRACT
A key problem in methods that automatically generate terrain which incorporate game level designs is a lack of quantitative measures that capture common game design elements. In this paper, we investigate a set of graph-connectivity and space-based metrics which can be used to classify area types that are commonly found in video game terrains. We evaluate the significance of each metric in differentiating area types by taking samples from a set of existing game levels with a known set of areas. Lastly, we demonstrate the potential of the metric set by creating classifiers that attempt to determine an areas' type based on its set of metrics.

Keywords
Procedural Content, Game Terrain, Virtual Terrain, Isovist, Graph-Connectivity

INTRODUCTION
Development of automated approaches for the design of game terrain is hindered due to there being no defined set of measures that quantitatively measure the suitability of a section of terrain for a particular purpose. Existing terrain generation techniques typically focus on aesthetic qualities rather than level design, while many level design generation techniques are not focused on terrain generation.

Many level design generation techniques work by restricting the problem to two dimensions. Such work includes Ashlock *et al.*'s (2011) method of generating maze-like levels, which uses a genetic algorithm (GA) to generate incorporate desired paths during level generation. Dormans (2010) and van der Linden & Bidarra (2013) took level design generation further and use mission/action graphs along with generative grammars to generate levels which fit a desired mission. This produces levels in such a way that the player is forced to experience game events in a desired order. Hartsook *et al.* (2011) introduced a similar method using a GA to generate Role Playing Game (RPG) levels,

Proceedings of DiGRA 2011 Conference: Think Design Play.

which, like Dormans' and van der Linden & Bidarra's approaches, forced the player to experience story events in a chronological order.

Some work has been done towards integrating game level design into virtual terrain by incorporating simple design elements into the terrain generation process. Olsen (2004) developed a method of generating game terrains using an efficient erosion algorithm. This created large, flat sections of terrain on which to place buildings and other game objects. This approach was designed around the real-time strategy (RTS) game genre and has limited use as a generic approach. Other work performed by Frade *et al*. (2010) also focuses on generating terrain with a flat area of desired size. Togelius *et al*. (2010) introduced a method using a multi-objective GA to generate RTS maps with level design elements including placement of resources, height advantages, and map symmetry. The downsides to this approach are the simple terrain representation, which results in unrealistic and un-detailed terrain, and their focus on the RTS genre, which prevents this from being a more generic approach. Smelik *et al*. (2011) proposed a framework where designers can place constraints within a virtual environment. These constraints interact with the environment as it is built, maintaining the requirements enforced by the constraint (such as enforcing line-of-sight between two areas).

These techniques are a start to generating complex three dimensional level designs, but there are still many game environment design elements to be considered. Hullett & Whitehead (2010) describe ten game design patterns commonly found in first-person shooter (FPS) game levels, including hidden areas, open areas, vantage points, choke points, and strongholds. These design elements have largely been ignored in procedural level generation and this is perhaps due to being unable to automatically identify these elements within a virtual environment. This article presents a collection of graph connectivity and space-based metrics, which we demonstrate are capable of differentiating game design elements in a 3D terrain environment. To demonstrate the effectiveness of these metrics for this purpose, we use three different classifiers to classify a dataset of area types described by Hullett & Whitehead. Only five of the design elements specified in Hullett & Whitehead were used in this research due to our approach classifying area types based on their physical properties, where five of the ten design elements described in Hullett & Whitehead were not types of areas.

The rest of this paper is organised as follows: The chosen metrics are detailed in "Preliminary Concepts", followed by the results of attribute analysis and classification from three types of classifiers in "Experiments and Results". Lastly a discussion of the results and the conclusion are presented in "Conclusion".

PRELIMINARY CONCEPTS

Our research uses a collection of space and graph-connectivity metrics to represent area types. This section describes what an isovist is, how it is generated, and lists the isovist and graph-connectivity metrics which are used in this research.

Isovist Metrics

An isovist (Benedikt, 1979) is the non-occluded space that is visible from a given location. Properties of an isovist are used in architecture to aid in building an environmental design. Notably, isovists have been used in the serious game "Supervisor" (van Bilson & Poleman, 2009), which is used to train oil rig supervisors. [USED FOR WHAT]

Isovists are commonly represented as a collection of vectors, called radials, which originate from the given location. The magnitude of an isovist's radial is equal to the distance of the nearest occluding obstacle along the radial's direction from the isovist's origin and is capped at a maximum view distance.

We used 3D isovists represented as 1024 radial vectors facing in random directions from the isovist's origin. Figure 1 shows an example 3D isovist of 1024 radials, with some radials intersecting walls and others reaching maximum view distance.

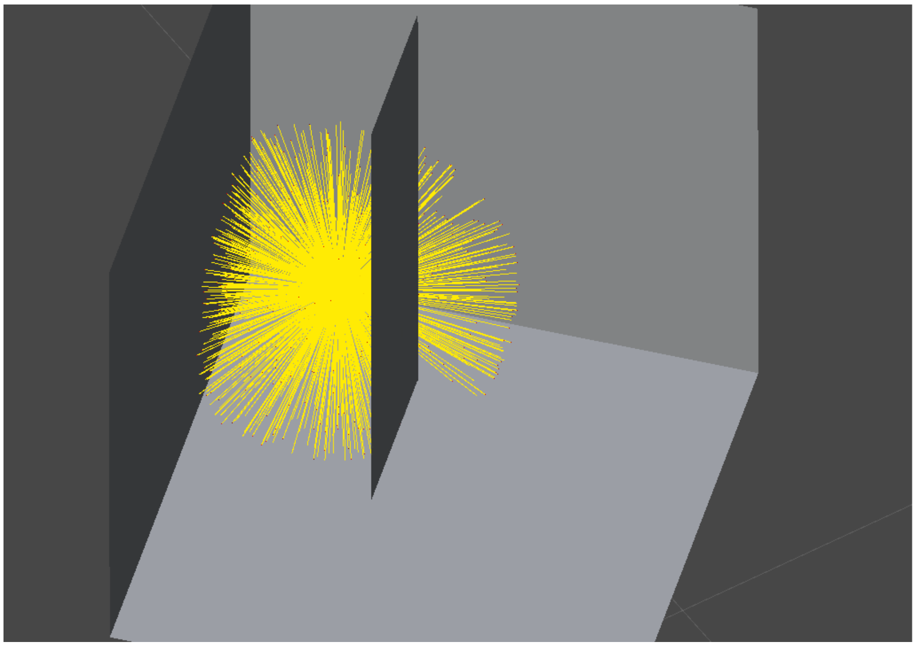

Figure 1. A 3D isovist with 1024 radials. Some of the isovist's radials are hitting occluding obstacles, and others reach maximum view distance.

Using this representation there are a number of measures which can be calculated, including those defined in Benedikt (1979) and Conroy-Dalton & Dalton (2001). All of the isovist metrics we use are included in Benedikt (1979) and Conroy-Dalton & Dalton (2001) with the exception of Area Size, which we obtain as a graph node property. A complete list of the isovist metrics used in this research is shown in Table 1.

Isovist Attribute Name	Isovist Attribute Description
Area Size	Surface area of flat terrain around a given location.
Isovist Average Radial Length	Average distance from a given location to its isovist's perimeter.
Isovist Dispersion	The difference between the values of the mean and the standard deviation of the isovist's radial lengths.
Isovist Drift 3D	The distance between the isovist's origin and the isovist's centre of gravity.
Isovist Drift 2D	Same as Drift 3D but ignores the Y (Up) axis.
Isovist Maximum Radial Length	Maximum distance from a given location to its isovist's perimeter.
Isovist Minimum Radial Length	Minimum distance from a given location to its isovist's perimeter.
Isovist Skewness	Skewness of the distance from a given location to its isovist's perimeter.
Isovist Sphericality	How well the isovist volume approximates a sphere.
Isovist Standard Deviation	Standard deviation of distance from a given location to isovist perimeter.
Isovist Variance	Variation in distance from a given location to its isovist's perimeter.
Isovist Volume	Area that can be seen (is not occluded) from a given location.

Table 1. A list of the isovist measures used in this research.

Graph Attribute Name	Graph Attribute Description
Degree Centrality	Number of edges connected to a node.
Eigenvector Centrality	The influence of the node in the graph based on its degree centrality and the degree centrality of its neighbours, and their neighbours, etc.
Betweeness Centrality	A measure corresponding to the number of shortest paths a node belongs to.
Closeness Centrality	A measure of how close a node is to every other node based on the shortest paths between them.

Table 2. A list of the graph connectivity measures used in this research.

Graph Connectivity Metrics

We use graphs to represent game levels as a collection of areas (nodes), possibly where game events occur, connected by paths (edges) which lead to new areas. Graph-connectivity metrics can then be used to estimate how likely it is an area will be visited by the player and how isolated the area is. The metrics we focus on are node centrality measures including, degree, eigenvector, betweeness, and closeness. Degree centrality for a particular node is defined as the number of nodes that node is connected to. As such, this corresponds to the number of areas that are accessible from a particular area. Eigenvector centrality is a measure of how important a node is based on how many nodes it is connected to and how important they are. Betweeness centrality is a measure of how much traffic a node is expected to receive in relation to each other node. Closeness centrality is a measure of how close a node is to every other node. These metrics are also listed in Table 1. Descriptions of these measures and how they are calculated can be found in Networks: An Introduction (Newman, 2010).

EXPERIMENTS AND RESULTS

Our research attempted to discover a collection of metrics that can be used to accurately classify an area type. To do this, a series of isovist metrics, commonly used in spatial analysis, were explored as they can capture spatial qualities of desired area types. Well known graph connectivity metrics were also explored, as graphs can represent the environment as a set of connected areas, not unlike a game level. This allows for identification of areas that are, for example, situated on commonly traversed paths or on paths that receives little traffic.

In order to discover how useful our metrics are at classifying area types we performed two types of analysis: attribute evaluation, and classification. Attribute evaluation involved using an algorithm to discover which of the attributes were most useful in classifying area types, while the classification analysis involved running a dataset through three types of classifiers to determine the accuracy of classification using our set of metrics. The rest of this section describes how our dataset was generated ("Generating the Dataset"), the results from attribute evaluation ("Analysis of Attribute Evaluation"), and the results from the classification analysis ("Analysis of Classification Results").

Generating the Dataset

Our data set consisted of 70 area instances, identified from a set of publically available terrain maps for the video game "Savage: The Battle for Newerth" (2003). Each instance contained values for 17 attributes. The first attribute was the type of area assigned to the instance, while the other sixteen attributes were the isovist and graph connectivity metrics displayed in Table 1 and Table 2. The following steps give an overview of the process used to generate our dataset.

1. Find existing game levels which contain the desired area types.

2. Generate a graph for the game level, with nodes for each area and edges for each path.

3. Extract isovist and graph measures from the desired area types in the game levels.

Step 1 involved gathering existing "Savage" maps which contained the area types we used in this research, which are listed in Table 3. During step 2, a graph was generated, which captured the physical layout and connectivity of the game level, as shown in Figure 2. A node was added to the graph for each area of the game level (a cross-section of paths or dead-end of a path) and an edge was added between nodes whose corresponding areas were connected by a physically traversable path. In step 3, an isovist was generated for each area type manually identified in the selected map. Each isovist was placed 50 units above ground level as 50 units was designated to be the eye height of a game character. Attribute values were then extracted from the area's graph node and isovist to fill an instance of the dataset.

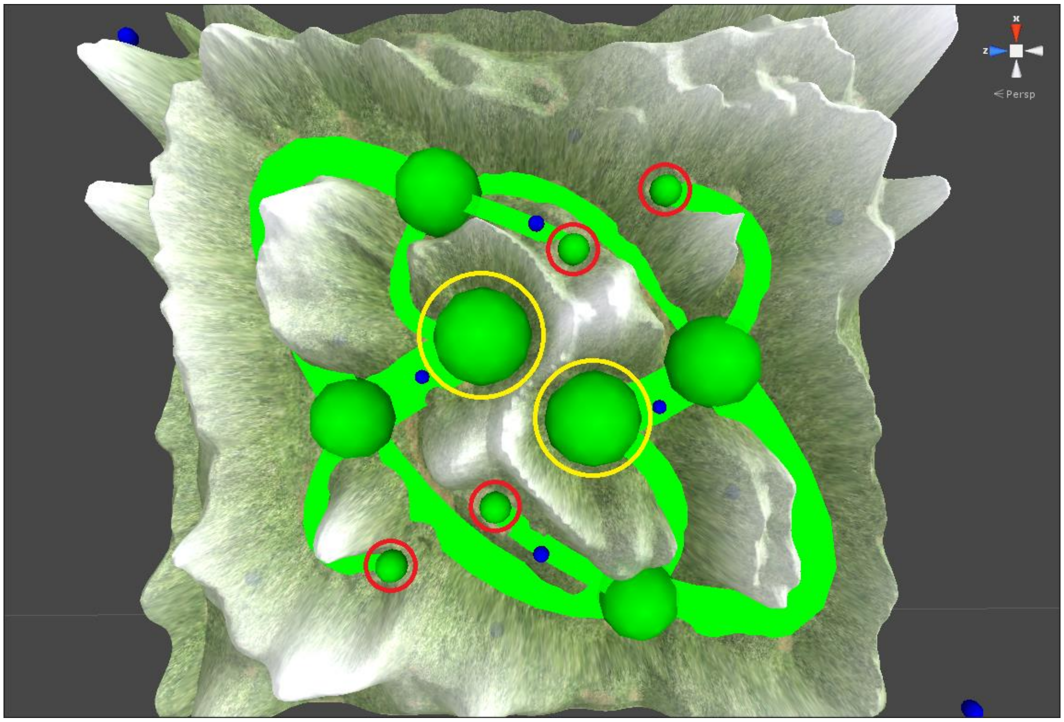

Figure 2. A screen shot of a player made level for the video game "Savage: The Battle for Newerth". The green spheres and lines show the physical layout of the level. The spheres circled in red were designated "vantage points", while the spheres circled in yellow were designated "Strongholds".

The process of selecting areas from existing levels and manually determining their area type, like most manual processes, is a subjective process. To make this process less subjective, areas were manually classified by comparing their physical attributes with the descriptions presented in Hullett & Whitehead (2010). As an example, Hullett & Whitehead's description of a "Sniper Location", which we label a "Vantage Point" as it is less genre specific, states that it is 1) an elevated position, and 2) overlooks a portion of the level. Figure 3 shows an example of an area which was manually classified as a vantage point. From this figure it can be seen that the vantage point is elevated and overlooks a large stretch of the game level. Therefore this area met the criteria of a vantage point.

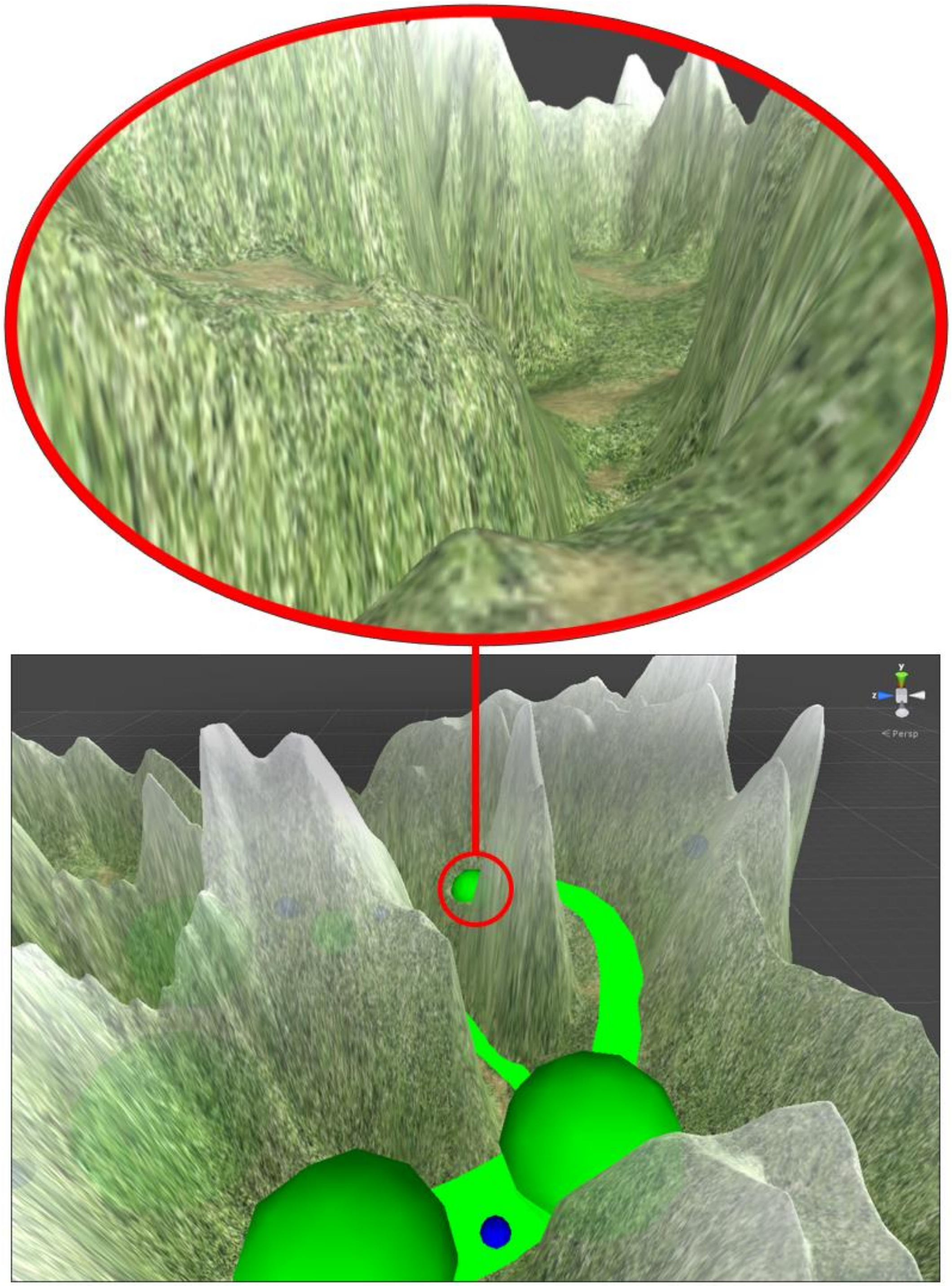

Figure 3. An area manually identified as a vantage point with a screenshot of the players vision from that location.

Due to some area types being more common than others the number of instances for each area type is not even, but a minimum of ten instances for each type was obtained. The different area types and, their number of instances, are listed in Table 3.

Area Type	Number of Instances
Hidden Area	11
Open Area	15
Vantage Point	19
Choke Point	15
Stronghold	10

Table 3. A list of the area types, used in this research, and their number of instances in the dataset.

Analysis of Attribute Evaluation

The dataset is run through an attribute evaluator to determine which attributes are most influential in the classification process. We use the tool "Weka" (2015), a collection of machine learning algorithms written in Java, to perform attribute evaluation. The attribute evaluation process in Weka requires an attribute evaluation method and search method to be selected. For these tests we selected the InfoGain attribute evaluator and the Ranker search method. After Weka processed the dataset it output the list of attributes in order of the most useful to least useful. For each attribute Weka also output an InfoGain factor, a value based on the entropy between the attribute and each area type, to show which attributes have a greater probability of accurately classifying the area type. These results are displayed in Table 4.

InfoGain Factor	Attribute
1.144	Area Size
0.918	Degree Centrality
0.750	Isovist Sphericality
0.692	Eigenvector Centrality
0.692	Isovist Dispersion
0.626	Betweeness Centrality
0.455	Isovist Skewness
0.407	Isovist Volume
0.403	Isovist Average Radial Length
0.403	Closeness Centrality
0.379	Isovist Drift 3D
0.375	Isovist Standard Deviation
0.375	Isovist Variance
0.322	Isovist Drift 2D
0	Isovist Maximum Radial Length
0	Isovist Minimum Radial Length

Table 4. An ordered list of attributes used in the classification of area types. The list is ordered from most influential attributes to the least influential, as denoted by their InfoGain factor.

Table 4 shows that the most influential attribute is the area size, this was expected to be an important attribute as one of the most common descriptors of an area type is its size. The isovist volume was expected to have a high influence since it indicates how much visibility the area has, which is a key descriptor of some area types, but this attribute was ranked eighth. The minimum and maximum isovist radial lengths were ranked last with an InfoGain factor of 0. This was expected as every area in the data set has an equal value for both of these attributes, with the minimum value being the distance of the radials facing directly down (since each isovist was a set distance above ground level), and the maximum distance being radials facing directly up, which all reach maximum view distance. Although these metrics were not useful in our experiments, it is important to list them here as they may play a more definitive role in different environments, such as indoor game levels.

Analysis of Classification Results

This section displays the results from running the dataset through three types classifiers, chosen to determine if one type of classifier out-performed others (e.g. perhaps decision trees perform better than black-box AI for this task). These results were obtained by using Weka to generate each classifier using the data set and observing how accurately each one classified area types. The three classifiers included the J48 decision tree, the Naïve Bayes classifier, and a multilayer perceptron.

J48 Decision Tree

In Weka, J48 is a method of generating C4.5 decision tree. The generated tree can either be pruned or unpruned but was pruned in this experiment. Since the dataset was not large enough to separate into a training set and test set, 10 fold cross validation was used to obtain a better prediction of classifier accuracy. Weka offers a number of adjustable parameters when generating the J48 classifier, but this research just used the standard, default settings. A summary of the results are displayed in Table 5, and the confusion matrix is shown in Table 6.

Correctly Classified Instances	58	82.8571%
Incorrectly Classified Instances	12	17.1429%
Kappa Statistic	0.7832	
Mean Absolute Error	0.0759	
Root Mean Absolute Squared Error	0.2566	
Relative Absolute Error	23.962%	
Root Relative Squared Error	64.4569%	
Total Number of Instances	70	

Table 5. A summary of the results from running the dataset through Weka's J48 decision tree classifier.

Hidden Area	Open Area	Vantage Point	Choke Point	Stronghold	<-Classified As
9	0	1	0	1	Hidden Area
0	13	0	0	2	Open Area
1	0	16	1	1	Vantage Point
0	1	2	12	0	Choke Point
0	2	0	0	8	Stronghold

Table 6. The confusion matrix generated from running the dataset through Weka's J48 decision tree classifier.

This classifier worked well with a predicted accuracy of 82.8571%. The confusion matrix shows how many areas of each type were correctly and incorrectly classified. The results show that 12 out of 70 instances were misclassified. This was not unexpected as one area type may have many similar properties to areas of differing types.

Naïve Bayes

Naïve Bayes is a common probabilistic classifier. As with J48, 10 fold cross validation was used when running the Naïve Bayes classifier with Weka's default settings. The summary of results is shown in Table 7 and the confusion matrix is shown in Table 8.

Correctly Classified Instances	60	85.7143%
Incorrectly Classified Instances	10	14.2857%
Kappa Statistic	0.8194	
Mean Absolute Error	0.0633	
Root Mean Absolute Squared Error	0.2406	
Relative Absolute Error	19.9891%	
Root Relative Squared Error	60.4479%	
Total Number of Instances	70	

Table 7. A summary of the results from running the dataset through Weka's Naïve Bayes probabilistic classifier.

Hidden Area	Open Area	Vantage Point	Choke Point	Stronghold	<-Classified As
10	0	1	0	0	Hidden Area
0	13	0	0	2	Open Area
2	0	16	1	0	Vantage Point
0	1	0	14	0	Choke Point
0	3	0	0	7	Stronghold

Table 8. The confusion matrix generated from running the dataset through Weka's Naïve Bayes probabilistic classifier.

This classifier output a slightly higher accuracy than J48, achieving 85.7143% accuracy, correctly classifying 60 out of the 70 instances.

Multilayer Perceptron

The multilayer perceptron is an artificial neural network (ANN) which uses back propagation to identify instances. The ANN used in this research contained 16 input nodes (one for each attribute), a single hidden layer of 10 nodes, and four output nodes (one for each area type). As with the previous two classifiers, 10 fold cross validation and the Weka default settings were used. The summary of the results are shown in Table 9 and the confusion matrix is shown in Table 10.

Correctly Classified Instances	58	82.8571%
Incorrectly Classified Instances	12	17.1429%
Kappa Statistic	0.7832	
Mean Absolute Error	0.0759	
Root Mean Absolute Squared Error	0.2566	
Relative Absolute Error	23.962%	
Root Relative Squared Error	64.4569%	
Total Number of Instances	70	

Table 9. A summary of the results from running the dataset through Weka's Multilayer Perceptron ANN classifier.

Hidden Area	Open Area	Vantage Point	Choke Point	Stronghold	<-Classified As
10	0	1	0	0	Hidden Area
0	11	0	0	4	Open Area
1	0	17	1	0	Vantage Point
1	0	2	12	0	Choke Point
0	2	0	0	8	Stronghold

Table 10. The confusion matrix generated from running the dataset through Weka's Multilayer Perceptron ANN classifier.

The ANN had equal accuracy to the J48 decision tree classifier, achieving a predicted accuracy of 82.8571%. This classifier performed similarly to the previous two classifiers. One notable pattern among the results is that Open Areas, although classified correctly on most accounts, were only ever mistaken as strongholds and vice versa. This was not unexpected as Open Areas and Strongholds are similar areas and some may even be suitable as either.

CONCLUSION

In this paper we have investigated the impact of a set of graph-based metrics and isovist metrics to differentiate between terrain area types that are commonly found in video games. Experiments resulted in a ranking of the metrics by information gain, and when the metrics were used to train a set classifiers, the resulting accuracy, using 10-fold cross-validation, ranged between 82-85%, depending on the type of classifier used. This shows that these metrics can be used to differentiate between the area types explored in this research, which is an indication of their effectiveness at capturing important area defining qualities in general.

Future work may include the use of a larger data set so that both a training and test set may be used. Another possible area of exploration includes the use of an abstaining classifier, which can abstain from classifying an area if it is not similar to any of the known area types.

BIBLIOGRAPHY

Ashlock, D., Lee, C., McGuinness, C. (2011) “Search-Based Procedural Generation of Maze-Like Levels”. Computational Intelligence and AI in Games, IEEE Transactions on 3, 260-273.

Benedikt, M.L. (1979) “To take hold of space: Isovist fields”. Environment and Planning B: Planning and Design. 6, 1, 47-65.

Conroy-Dalton, R., and Dalton, N. (2001) “OmniVista:an application for isovist field and path analysis”. Presented at 3rd International Space Syntax Symposium.

Dormans, J. (2010). “Adventures in level design: generating missions and spaces for action adventure games”. Proceedings of the 2010 Workshop on Procedural Content Generation in Games. ACM, Monterey, California.

Frade, M., de Vega, F.F. and Cotta, C. (2010) “Evolution of artificial terrains for video games based on accessibility”. In Applications of evolutionary computation. Springer Berlin Heidelberg. 90-99.

Hartsook, K., Zook, A., Das, S., Riedl, M.O. (2011) “Toward supporting stories with procedurally generated game worlds”. Computational Intelligence and Games (CIG), 2011 IEEE Conference on.

Hullett, K. and Whitehead, J. (2010) “Design Patterns in FPS Levels”. In Proceedings of the 5th International Conference on the Foundations of Digital Games. 78-85.

Newman, M. (2010) “Networks: An Introduction”. Oxford University.

Olsen, J. (2004) “Realtime procedural terrain generation”. Technical Report. University of Southern Denmark.

S2 Games (2003) Savage: The Battle for Newerth [PC Computer, Online Game] Desura. Rohhnert Park USA.

Smelik, R. Galka, K. de Kraker, K. L. Kuijper, F. Bidarra, R. (2011) “Semantic constraints for procedural generation of virtual worlds”. Proceedings of the 2nd International Workshop on Procedural Content Generation in Games. ACM.

Togelius, J. Preuss, M. and Yannakakis, G. (2010) “Towards multiobjective procedural map generation”. Proceedings of the 2010 Workshop on Procedural Content Generation in Games. 1-8.

University of Waikato (2015) Weka [PC Computer] University of Waikato. Hamilton New Zealand.

van Bilsen, A. and Poleman, R. (2009) “3D Visibility Analysis in Virtual Worlds: The Case of Supervisor”. In Proceedings of Construction Applications of Virtual Reallity. 5, 6, 267-278.

van der Linden, R.L., R., Bidarra, R. (2013) “Designing procedurally generated levels”. Proc. 2nd Workshop Artif. Intell. Game Design Process 41-47.