ICA Workshop on Generalization and Multiple representation; 20-21 August 2004 - Leicester

3D building recognition using artificial neural network

Jagdish Lal, Liqiu Meng
Technical University of Munich
Arcisstr.21 / 80333 München / Germany

Email: jagdish@bv.tum.de, meng@bv.tum.de
http://www.carto-tum.de

Abstract

What a three dimensional (3D) building presents is often very complex visual stimuli of multidimensional semantic information. Developing a computational model for its recognition is difficult. Neural network (**NN**) algorithms have been widely used for **pattern recognition** because of their abilities of self-organization and parallel processing characterized by robustness and fault tolerance. They can learn by adapting their synaptic weights to changes in the surrounding environment, handle imprecise, incomplete, fuzzy as well as noisy information and generalize from known or unknown tasks. **Artificial Neural Networks** (ANN) are an attempt to mimic some of these characteristics. In this paper, this idea is extended to the automatic recognition 3D buildings, which is required by the subsequent generalization. A hierarchical approach was applied to various types of buildings ranging from simple (single) buildings to complex (group of simple buildings) buildings.

Keywords — Artificial Neural Networks, Generalization, building recognition

1. Introduction

Generalization is the process of creating a legible map at a given scale from a large scale [Ruas, 1996]. It is done in such a manner that the character or essence of the original features is retained at successively smaller scales. In 3D city model, which consists mainly of buildings and roads, their structure recognition, a prerequisite for generalization [Lal & Meng, 2001] requires to be studied in more details. Though it is very difficult to describe all kinds of buildings, most buildings belong to one of the types summarized in figure 1. These buildings can be represented as a combination of several simple building parts which in turn are represented by sets of points, lines and regions. Further these buildings can be divided into different groups based upon their roof styles as follow:

Simple buildings: These buildings correspond to different roof styles, e.g., pyramid, gable, hip etc.
Complex buildings:

- Buildings which have a roof as a combination of the same or different roof styles, e.g., cross - gable, gable-lean etc. as shown in figure 1.
- Building as a group of small buildings joined together forming a special shape, e.g. closed square, bridge and cross.

Structure recognition of these buildings can be divided into three levels, namely micro, meso and macro level that respectively stand for individual building, buildings in neighborhood and building blocks. An attempt has been made here to recognize different building types using Artificial Neural Network (ANN) technique. Following paragraphs give an introduction to ANN followed by recognition of buildings.

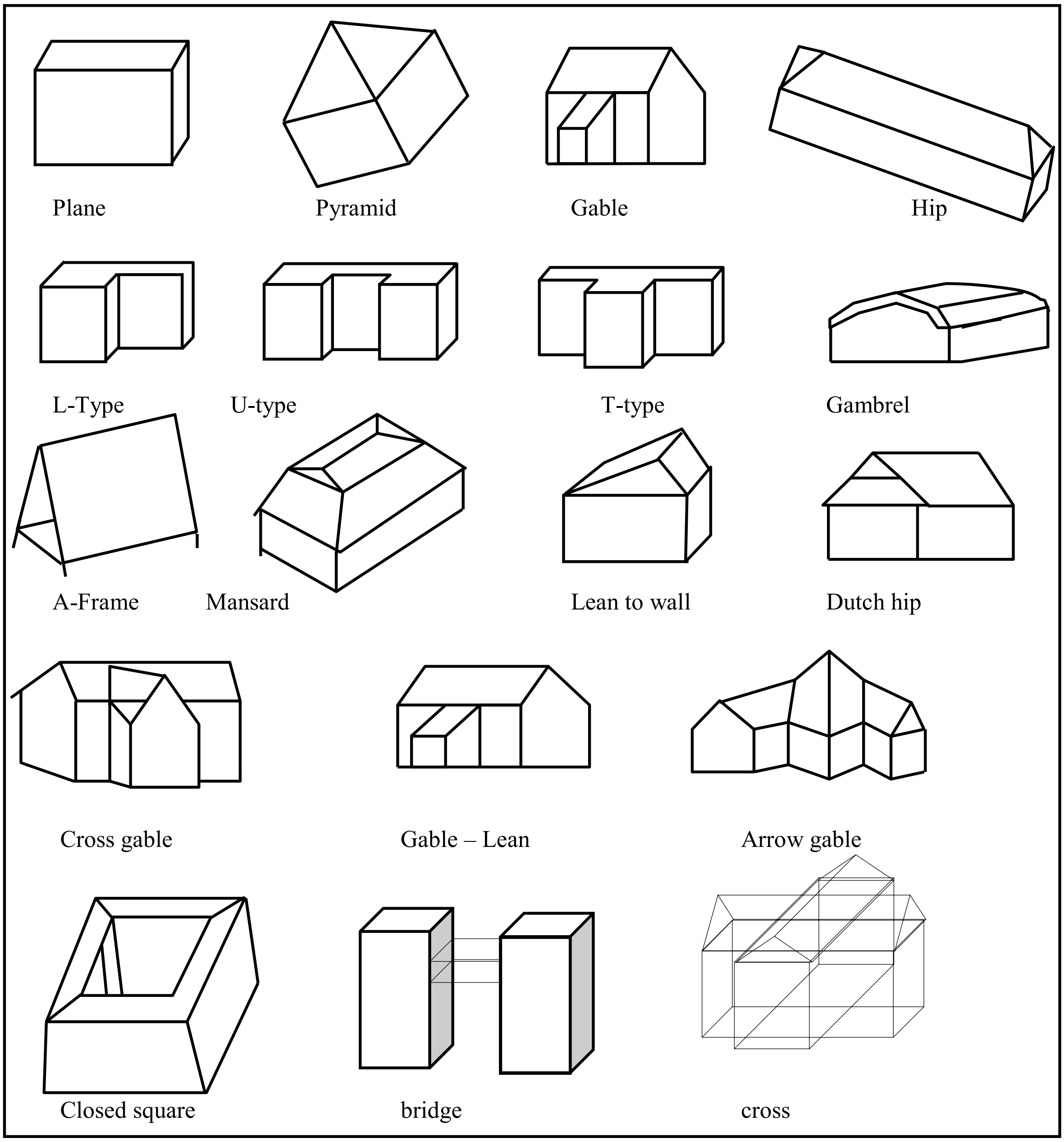

Figure 1: Different building types

2. ARTIFICIAL NEURAL NETWORK

2.1 Concept: In simple terms, ANNs are collections of mathematical models that emulate some of the observed properties of biological nerve systems and draw on the analogies of adaptive biological learning [Haykin, 1994]. The key element of the ANN paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements that are analogous to neurons and are tied together with weighted connections that are analogous to synapses.

2.2 Structure of an ANN: A typical ANN has N inputs and one output. The input layer is composed not of full neurons, but rather consists simply of the values in a data record that constitutes inputs to the next layer of neurons. The next layer is called a hidden layer and there may be several hidden layers. The final layer is the output layer, where there is one node for each class. A single sweep forward through the network results in the assignment of a value to each output node, and the record is assigned to whichever class's node had the highest value (see figure 2).

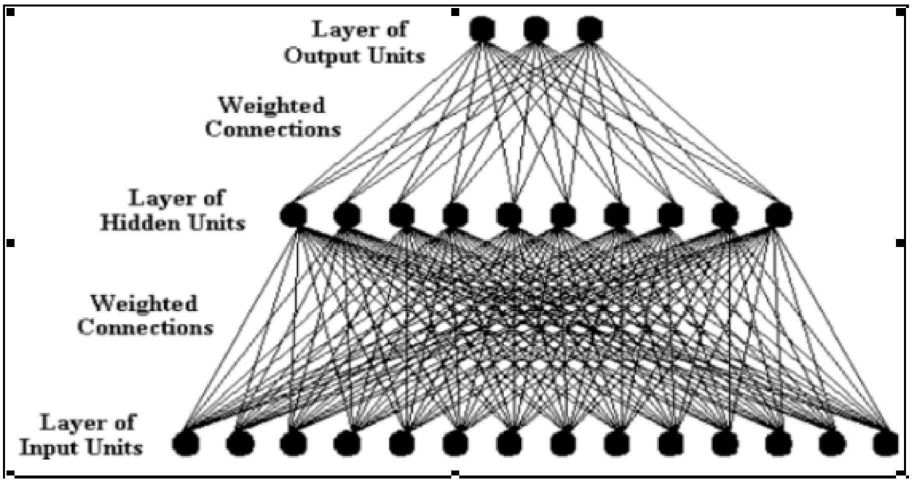

Figure 2: ANN structure

A successful example of ANN applied to an optical character recognition (OCR) by [Neuro solution, 2002] is shown in figure 3. The original document is scanned into the computer and saved as an image. The OCR software breaks the image into sub-images, each containing a single character. The sub-images are then translated from an image format into a binary format, where each 0 and 1 represents an individual pixel of the sub-image. The binary data is then fed into a neural network that has been trained to make the association between the character image data and a numeric value that corresponds to the character. The output from the neural network is then translated into ASCII text and saved as a file.

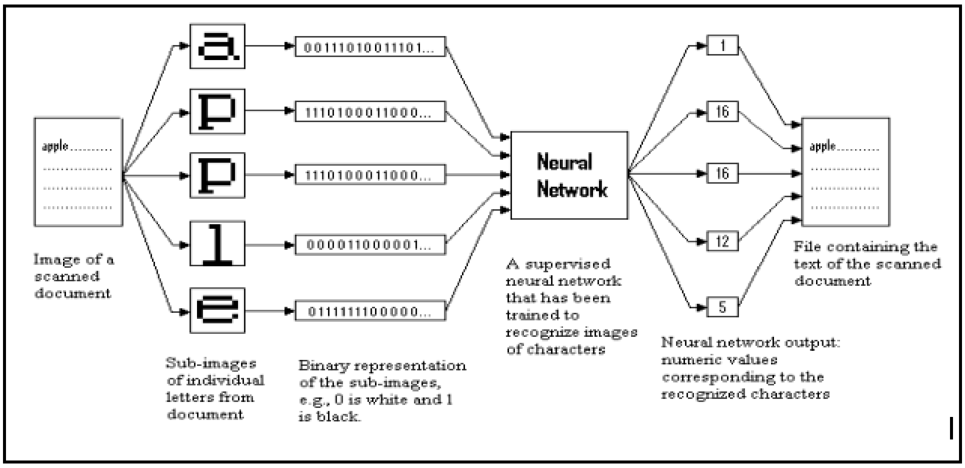

Figure 3 : ANN Example (source: Neuro solution, 2002)

2.3 Recognition of 3D buildings: One of the important representations of a building is using "B-rep" having layered description of a geometric object. The first layer contains zero-dimensional faces (vertices), the second layer one-dimensional faces and so on. Using this representation, solids can be described unambiguously by describing their surface and topologically orienting it such that it is possible to tell, at each point of the surface, on which side of the surface the interior of the solid lies. This includes a topological description of the connectivity and orientation of vertices, edges, and faces, and a geometric description for embedding these surface elements in space. Further the topological description specifies vertices, edges, and faces and indicates their incidences and adjacencies. The geometric description specifies, for example, the equations of the surfaces of which the faces are a subset. Most of the information available about a building using this representation will be used as the input to the neural network.
A 3D building consists of different characteristic parts, e.g. ground plan, roof or simple buildings. To recognize it, a hierarchical approach has been adopted. It involves Recognition of:

- Ground plan
- Roof type
- Simple building
- Complex building

A city model of Bonn is taken for recognition. It consists of approximately 400 different individual buildings ranging from simple to complex as shown in figure 4.

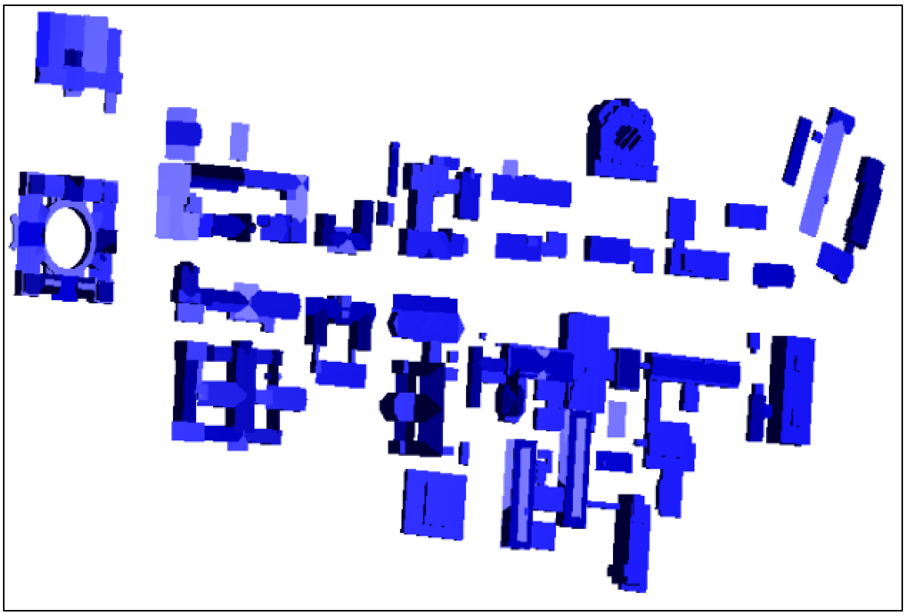

Figure 4: Nussalle area of City Bonn

Most of the buildings in figure 4 of the Nussalle area of the city are complex. It means they in turn are made of simple buildings belonging to one of the types shown in figure 5.

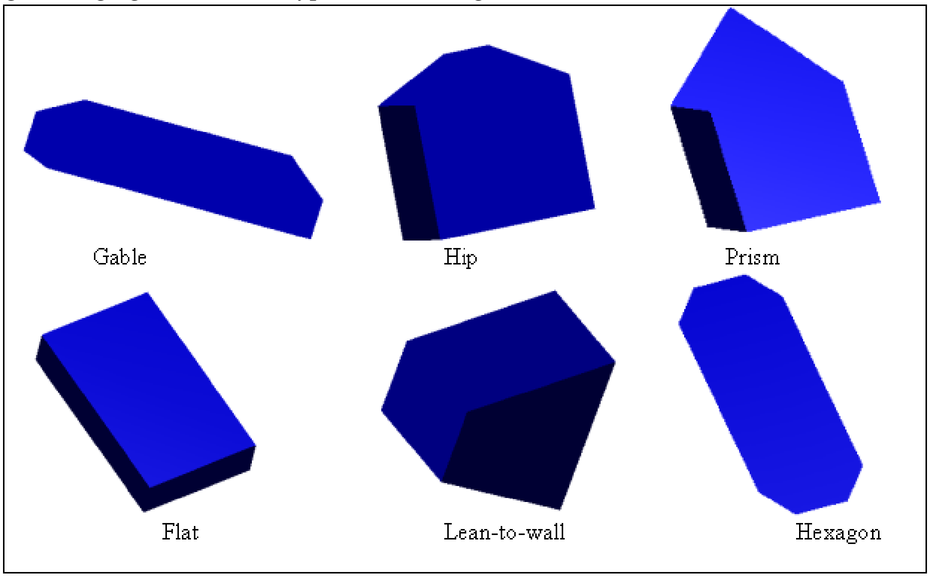

Figure 5: Building types in the test area

These buildings are named Gable, Hip, Prism, Flat, Lean-to-wall and Hexagon. A complex building is formed by a group of these simple buildings after their rotation, translation and scaling.
A hierarchical approach is applied to recognize their ground plan, roof style and then the building as a whole as result of one step will serve as input to the next step i.e. ground plan will act as inputs to the recognition of roof style and both ground plan and roof style will serve as input to the recognition of building.

3. RECOGNITION OF SIMPLE BUILDINGS

Our input data contains 250 different buildings of six types shown in figure 5. Out of them 100 buildings are selected for training the ANN and rest of the buildings are used for predictions. ANN trains itself using a given set of input but it is difficult to decide how large the input set should be for its perfect training before it gives acceptable results. Therefore an incremental approach of adding an input building is selected. In the beginning of the trial, only coordinates of the individual buildings are given as an input but the output was not up to the desired mark. Other parameters are added step by step until a satisfactory result is obtained. Table 1 shows the parameters selected as possible input to the ANN.

Sr. No	Input	Description
I	*ID*	*Primary key field (0-250)*
ii	*RoofType*	*(2-flat,3-gable,4-hip,5-prism,6-hexagon,7-lean-to-wall)*
iii	NoOfFaces	Total number of the faces of a building
iv	GroundType	(1-square or rectangular, 2-hexagon)
V	FaceAngle	Angle between the roof faces, if any
vi	*X,Y,Z*	*Coordinates of the building vertices*
vii	TotalVerteces	Total number of vertices of the building
	Output	Description
	Building Id	(1-flat,2-gable,3-hip,4-prism,5-hexagon,6-lean-to-wall)

Table 1: Possible parameters as input to NN

The parameters shown in black are the optimally tuned ones for the successful predictions and remaining parameters (italic & colored) are not used. Here the type of ground plan is results from 3.1 and 3.2. Figure 6 shows the window for entering the parameters. One parameter is added at a time and ANN is trained till the training is satisfactory. Certain parameters are also removed and ANN is again trained to check if these parameters are redundant.

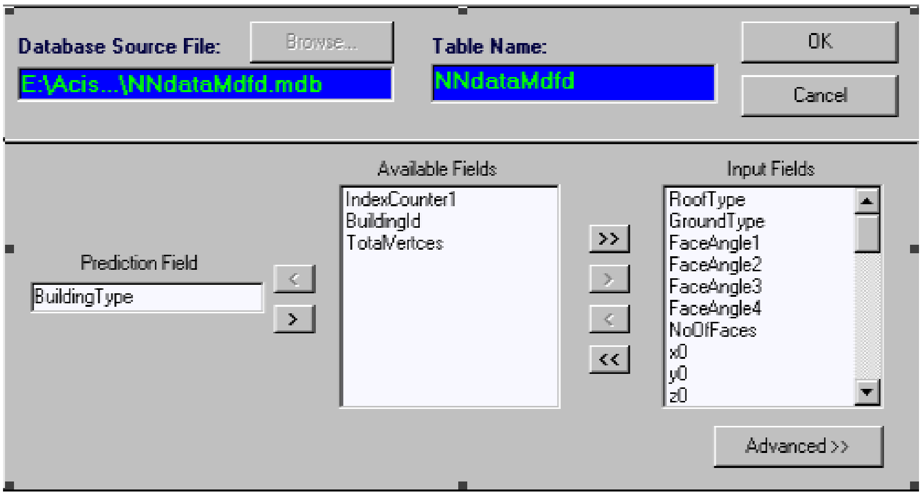

Figure 6: Input & output selection window

Neural Network Training: The training begins with all weights set to random numbers. For each data record, the predicted value is compared to the desired (actual) value and the weights are adjusted to move the prediction closer to the desired value. Many cycles are made through the entire set of training data with the weights being continually adjusted to produce more accurate predictions.

Learning of the network is controlled by setting three parameters viz. *learn rate, momentum and verify rate* as shown in *history graph* in figure 7. This graph shows a history of the prediction error achieved during the previous verify cycles. *Learn rate* and *momentum* are best set during training run but in the beginning, it is better to set *learn rate* always greater than *momentum*. If *learn rate* and *momentum* are set too low, the training will be very slow with a smooth, gradual improvement. If *learn rate* and *momentum* are set too high, the training will be very choppy, and chaotic. *Verify rate* determines how many training cycles are made before a verify cycle is run and is necessary to evaluate the current network and report the error.

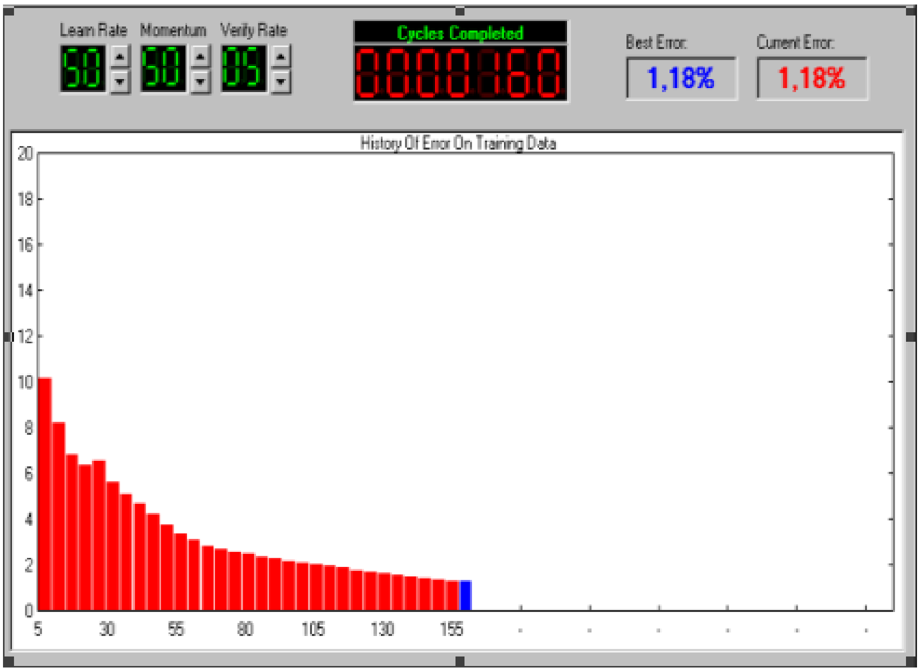

Figure 7: History graph

Scatter Graph*:* This provides both a numeric and graphical report showing the accuracy of network predictions (figure 8).

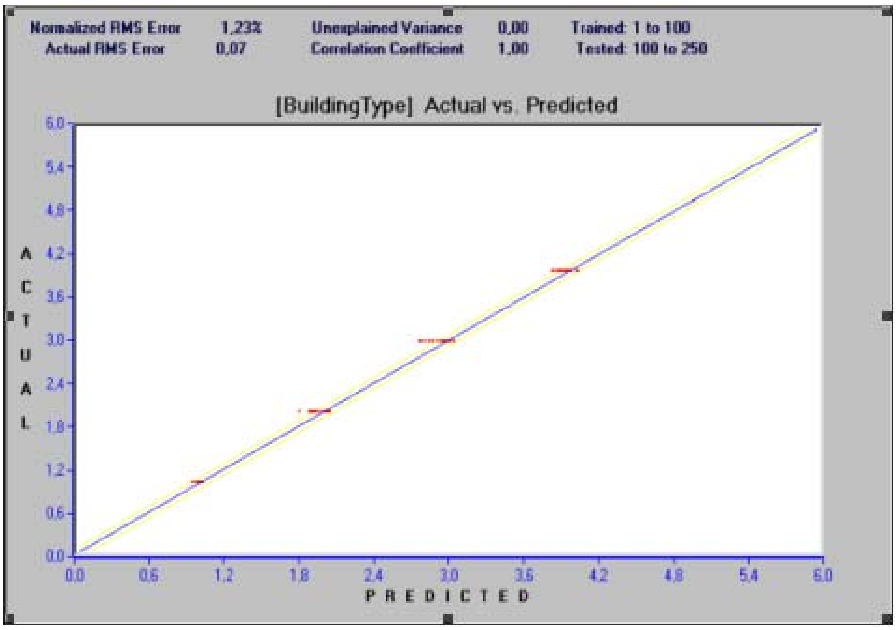

Figure 8: Scatter graph

The closer the scattering (red dots) is to the blue diagonal line, the more accurate are the predictions. The band shown by faint yellow lines indicates a certainty band, as defined by the RMS Error. Approximately 2/3 of the testing set should lie within this certainty band.

Predictions: Once the learning is over, prediction results can be seen immediately as shown in figure 9. It shows the given building ID (type of building), predicted ID and the difference between the twos.

BuildingType	Predicted	Difference	GroundType	TotalVertces	FaceAngle1	FaceAngle2	FaceAngle3	FaceAngle4
4	3,17	0,83	1	10	6869	6869	6361	6362
3	3,59	0,59	1	10	7901	7900	7847	7846
3	3,76	0,76	1	10	10461	10460	6207	25192
6	5,84	0,16	2	10	9089	6759	15880	0
1	1,03	0,03	1	8	0	0	0	0
6	5,84	0,16	2	10	8414	7456	15528	0
6	5,83	0,17	2	10	9543	6269	15586	0
1	1,03	0,03	1	8	0	0	0	0
1	1,03	0,03	1	8	0	0	0	0
3	3,31	0,31	1	10	2747	2749	2461	28938
2	2,07	0,07	1	10	14551	7260	7259	0
1	1,03	0,03	1	8	0	0	0	0
2	2,10	0,10	1	10	16868	7257	7273	0
2	2,07	0,07	1	10	14551	7260	7259	0
1	1,03	0,03	1	8	0	0	0	0
3	3,40	0,40	1	10	3746	3747	4233	27166
3	3,40	0,40	1	10	3746	3747	4233	27166
1	1,03	0,03	1	8	0	0	0	0
1	1,03	0,03	1	8	0	0	0	0
3	3,34	0,34	1	10	6418	6422	4084	27315
1	1,03	0,03	1	8	0	0	0	0
2	2,12	0,12	1	10	9957	4970	4986	0
2	2,15	0,15	1	10	8960	4472	4487	0
1	1,03	0,03	1	8	0	0	0	0
1	1,03	0,03	1	8	0	0	0	0
3	3,09	0,09	1	10	3909	3908	3665	3664
2	2,11	0,11	1	10	10468	5224	5243	0
2	2,12	0,12	1	10	9671	4894	4776	0
6	5,85	0,15	2	10	13684	5247	8436	0
2	2,12	0,12	1	10	18534	9270	9263	0
4	3,36	0,64	1	10	3830	3829	3699	27700
2	2,24	0,24	1	10	5964	2966	2966	0
3	3,44	0,44	1	10	7979	7978	7589	23810
2	2,12	0,12	1	10	19036	6170	6193	0
2	2,15	0,15	1	10	9057	4518	4538	0
3	3,55	0,55	1	10	9789	9789	8987	22412
5	4,92	0,08	1	12	0	0	0	0
3	3,10	0,10	1	10	3846	3846	3717	3717
2	2,12	0,12	1	10	10161	5072	5088	0
2	2,08	0,08	1	10	12063	6031	6032	0
3	3,38	0,38	1	10	7330	7330	6074	25325
3	3,41	0,41	1	10	4842	4842	5598	25801
2	2,10	0,10	1	10	17949	6732	6717	0
2	2,11	0,11	1	10	18290	6562	6546	0
3	3,44	0,44	1	10	8046	8046	7588	23811
3	3,55	0,55	1	10	9795	9789	9067	22332

Figure 9: Prediction results

It shows that prediction results are very good and close to the expectations. It completes the whole cycle of the learning and predicting of building types. As the complete process is described in details for the recognition of buildings, next recognition of roofs and ground plans will only describe it briefly.

3.1 Recognition of roofs: In recognition of roof, same data is taken but different set of parameters is chosen. Following parameters are selected as input:

Sr. No	Input	Description
i	NoOfRoofFaces	Total number of the faces of a roof
ii	FaceAngle	Angle between the roof faces, if any
iii	X,Y,Z	Coordinates of the building vertices
Output		Description
RoofType		(2-flat,3-gable,4-hip,5-prism,6-hexagon,7-lean-to-wall)

Table 2: Building parameters

After running the network following the same steps as described above, following results are obtained shown in figure 10

RoofType	Predicted	Difference	NoRoofFaces	FaceAngle1	FaceAngle2	FaceAngle3	FaceAngle4
4	4,00	0,00	4	7593	7588	6056	6050
2	2,00	0,00	1	0	0	0	0
3	3,11	0,11	2	10335	5151	5151	0
3	3,53	0,53	2	7691	3846	3845	0
3	3,38	0,38	2	7676	3830	3846	0
5	4,25	0,75	4	3329	3328	3840	3839
7	6,77	0,23	1	0	0	0	0
5	4,75	0,25	4	3329	3329	2514	2514
2	2,00	0,00	1	0	0	0	0
2	2,00	0,00	1	0	0	0	0
3	3,00	0,00	2	11150	5575	5575	0
2	2,00	0,00	1	0	0	0	0
5	5,03	0,03	4	6135	6139	6038	25361

Figure 10: Prediction results

Though the results are very good for most of the roof types but for roof type five (i.e. prism), there is little difference between actual and prediction. These results can be further improved by adding one more parameter.

3.2 Recognition of ground plan: The current data contains buildings with two types of ground types viz. Rectangular and hexagonal. To recognize them, only the coordinates of the buildings are used and results are obtains as shown in figure 11.

GroundType	Predicted	Difference	x0	y0	z0	x1	y1
1	1,000	0,000	1021	-61	-234	1007	-61
1	1,002	0,002	1015	-60	-135	1022	-60
1	1,001	0,001	1006	-60	-114	1006	-79
1	1,002	0,002	1031	-60	-113	1023	-60
1	1,001	0,001	1005	-60	-92	996	-60
1	1,001	0,001	1002	-60	-119	1002	-79
1	1,001	0,001	993	-60	-115	1000	-60
1	1,001	0,001	993	-60	-115	1000	-60
1	1,001	0,001	1010	-60	-84	1010	-66
1	1,001	0,001	1009	-59	-68	1036	-59
1	1,000	0,000	1031	-59	-102	1031	-67
1	1,002	0,002	1043	-61	-257	1035	-61
1	1,002	0,002	1031	-61	-276	1044	-61
1	1,000	0,000	1043	-60	-141	1043	-66
1	1,000	0,000	1070	-60	-157	1057	-60
1	1,001	0,001	1053	-60	-166	1062	-60

Figure 11: Prediction results

4. RECOGNITION OF COMPLEX BUILDINGS

Complex buildings are composed of group of simple buildings as these buildings are located in so close vicinity that the whole group forms a complex structure. Figure 12 shows one such complex building, which is, formed from more than 30 simple buildings shown in figure 2.

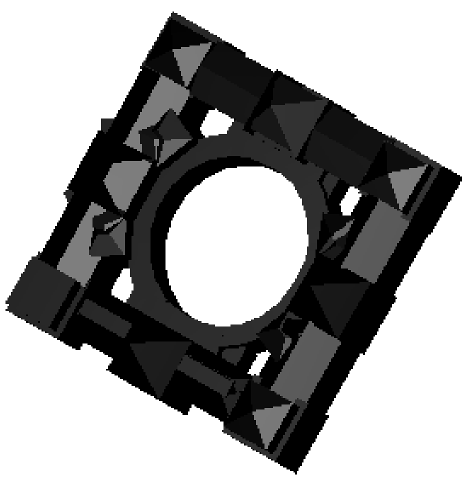

Figure 12: A Complex building

Recognition of such buildings is a tedious task as it involves the extraction of complete information of simple buildings joined to form these complex buildings. Fortunately complete information about the individual buildings is already available as described above in *recognition of simple buildings* but the difficult task still remains as how to get the identification of participating buildings. Well-known algorithms like Delaunay Triangulation have proved unsuccessful by taking centers of the simple buildings as its input. Since these buildings are of various length and shape, difficulty was faced in deciding the cut-off value of edge distance. Too small values gave more than one cluster of a complex building and too large vales gave cluster of simple buildings comprising complex building plus other nearby buildings.

A new algorithm is developed which takes care of the shortcoming of the above algorithm. It starts with a single building and finds its immediate neighbors. Once the immediate neighbors are found, next step involves the finding of immediate neighbors of these newly found neighbors and the process repeats till there is no more neighbor. It gives rise to the cluster of simple buildings forming a complex building. The process repeats itself for the next cluster till there is no more cluster left. A flowchart of the algorithm is shown in figure 13.

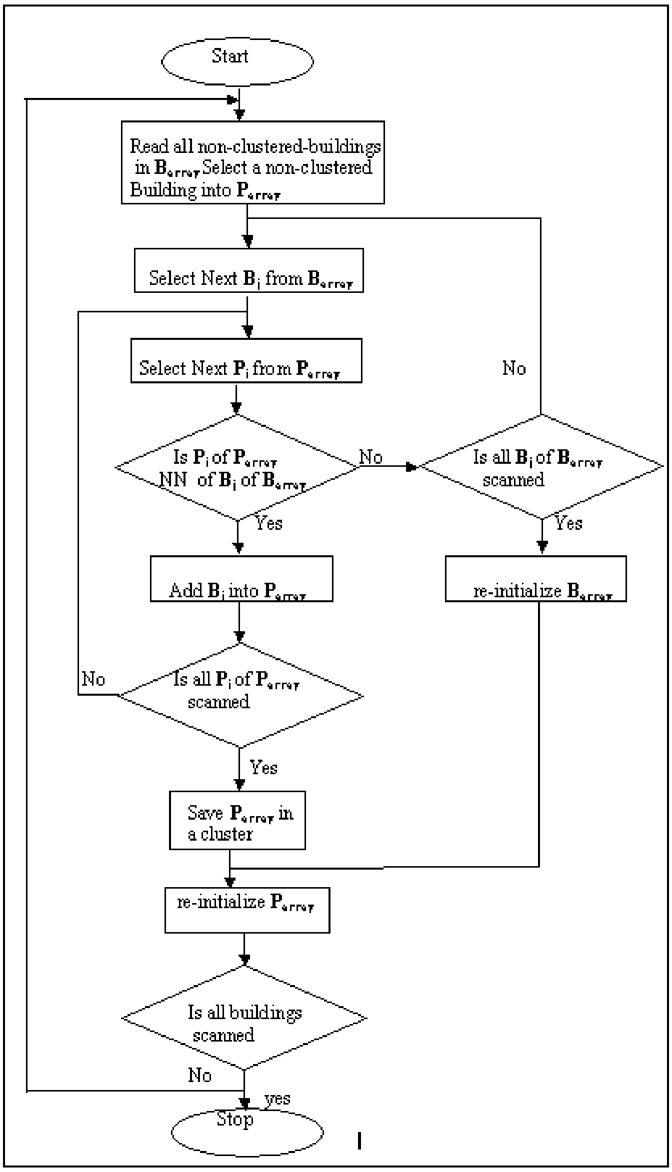

Figure 13: A Cluster algorithm

This algorithm is implemented in C++ and successfully applied to the building of Bonn shown in figure 2. After applying algorithm, figure 14 shows some of the important clusters obtained

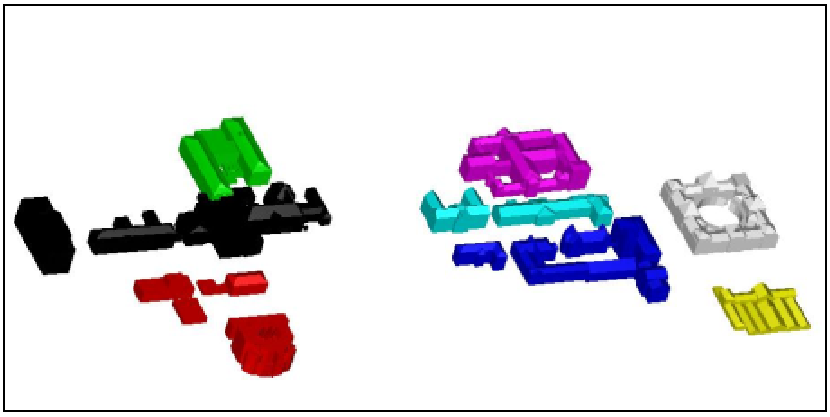

Figure 14: Clustered buildings

Each color represents a cluster that in turn represents a complex building. As the complete information of the participating buildings is known now, the next step is to remove the redundant information in them. For example, two building which are very close, the two intersecting faces (walls) have the same vertices, secondly, as only the outer surface is required for recognition, either face may not play any important role in their recognition and hence to be removed. It is done by applying convex hull to the entire complex buildings so that just the outer vertices are found. Therefore convex hull algorithm is applied and figure 15 shows a convex hull wire frame for one of the above building.

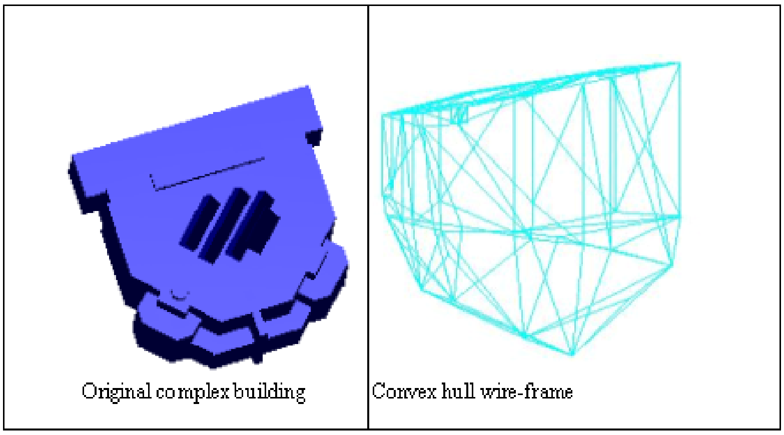

Figure 15: Convex hull

Once the convex hull is successfully applied, other parameters which are known now can be applied as input to ANN are:

- Topology
- Ground plan
- Roof heights
- Roof styles
- Roof face angles
- Building area
- Wall faces
- Roof faces
- Ground faces

These parameters are added step by step till the best possible result is found in a manner described above. Due to lack of sufficient data, it is not possible to use ANN for complex buildings but the process is exactly the same as described above.

5. Conclusion

In this paper, we discussed the recognition of 3D buildings using ANN – a prerequisite for the subsequent generalization process. A real 3D data set of Bonn has been used as the input to the ANN. This study has shown that ANN can be applied successfully, like pattern recognition [Nigrin, 1993], to building recognition ranging from simple to complex, scaled and transformed. This approach is also suitable when the input data is incomplete.

6. Acknowledge

This research is being carried out under the joint research project sponsored by German Natural Science Foundation, *Generalization of 3D Settlement Structures on the Basis of Scale Space and Structure Recognition,* between the Department of Cartography, Technical University of Munich (TUM), and the Institute of Photogrammetry and Cartography, University of German Armed Forces in Munich. During this work, we used a downloadable ANN software "NeuNet Pro" and would like to thank the owner of the company (www.cormachtech.com/newnet) for this excellent package.

7. References

Ruas, A. & Plazanet, C. (1996): Strategies for automated generalization. Proceedings 7th International Symposium on Spatial Data Handling. Delft (Netherlands), Vol.1, pp 6.1-6.18

Ruas, A. (1998): Constraint Analysis. Report DA2, 1998, Dept. of Geography, University of Zurich

Haykin, S. (1994): Neural Networks: A Comprehensive Foundation, NY: Macmillan.

Lal, J. & Meng, L. (2001): Rules and constraints for 3D generalization of urban area. Journal of Geographical Sciences, Acta Geographica Sinica, Vol.11 Supplement 2001, Beijing. 17-28

Neuro solution (2002): What is a Neural Network?

Nigrin, A. (1993): Neural Networks for Pattern Recognition, Cambridge, MA: The MIT Press.

Pinkus, A. (1999): Approximation theory of the MLP model in neural networks- Acta Numerica, 8, 143-196.

Zurada, J.M. (1992): Introduction To Artificial Neural Systems, Boston: PWS Publishing Company.