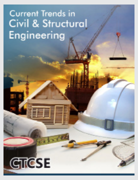

ISSN: 2643-6876

Current Trends in
Civil & Structural Engineering

DOI: 10.33552/CTCSE.2020.04.000599

Iris Publishers

Mini Review

Cost-Effective as-Built BIM Modelling Using 3D Point-Clouds and Photogrammetry

Stephen Oliver[1], Saleh Seyedzadeh[2], Farzad Pour Rahimian[1]*, Nashwan Dawood[1] and Sergio Rodriguez[1]

[1]*School of Computing, Engineering and Digital Technologies, Teesside University, Tees Valley, Middlesbrough, UK*

[2]*Department of Engineering, University of Strathclyde, Glasgow, UK*

***Corresponding author:** Farzad Pour Rahimian, School of Computing, Engineering and Digital Technologies, Teesside University, Tees Valley, Middlesbrough, UK.*

Received Date: December 20, 2019
Published Date: January 09, 2020

Abstract

The classification and translation of 3D point-clouds into virtual environments is a time consuming and often tedious process. While there are large crowdsourcing projects for segmenting general environments, there are no such projects with a focus on AEC industries. The nature of these industries makes industry-specific projects too esoteric for conventional approaches to data collation. However, in contrast to other projects, the built environment has a rich data set of BIM and other 3D models with mutable properties and implicitly embedded relationships which are ripe for exploitation. In this article, the readers will find discussion on contemporary applications of image processing, photogrammetry, BIM and artificial intelligence. They will also find discussions on the applications of artificial intelligence, photogrammetry in BIM and image processing that will likely be at the forefront of related research in the coming years.

Introduction

The aim of this paper is to propose a framework for the creation of a BIM model using the collected 3D point-cloud and a series of related datasets in the form of Linked Data [1]. This aim will be achieved by employing data mining and machine learning methods [2-4] for extracting the relevant information in creating meaningful features for recognition of building elements and semantic annotation. These identified components will be loaded as BIM entities. Diverse and extensive data can be obtained from data sets with seemingly low richness. The aim is to create a mechanism for producing and extending Linked Open Data (LOD) sets through machine- guided processing and where appropriate, self-propagation. The challenge is identifying which data can link otherwise foreign data sets and then identifying and extending previously traversed data. There are three essential steps

1) Collating appropriate and verified data sets whether strictly or implicitly assured.

2) Standardizing and aggregating data into a common schema.

3) Iteratively enriching existing data as new linking features are identified.

A key ambition reliant on these steps is the classification and translation of 3D point clouds from Unmanned Aerial Vehicles (UAVs). The classification potential for these is vast however, their density is prohibiting and the translation process challenging. There are four keys steps in initiating the processes.

1) Clustering points of the same entity into individual profiles.

2) Segmenting the components of the identified entities.

3) Identifying entity relationships.

4) Reduction of properties into the common schema and where spatial data is relevant, into a discrete parametric model.

These define both the foundational datasets and aid identification of a target location's probable solution space. For example, knowledge of spatial and material properties, and historically observed proximal entities enable mixed-method virtual reconstruction and verification of low-confidence classifications. Even higher-level properties such as architectural styles have profound implications for reducing the complexity of a classification problem.

Technical details

Bridging the gap between real and virtual worlds is no longer a matter of technological barriers but rather resource management. The tools necessary to link, translate and fuse mixed reality data exist in isolation, it is down to development teams to figure out how they may be combined to produce a practical utility under the constraints of funding and team capacity. Drone LiDAR scanning is long established for producing a discrete spatial mapping of the real world, which can be translated into virtual components. To a lesser extent, some tools such as Tango-enabled mobiles can produce and translate point-clouds to colored 3D meshes in near real-time. Pour Rahimian et al. [5] demonstrated translation between vector and discrete virtual environments and linking Cartesian and Barycentric coordinate systems. In a related project, [6] extended bidirectional linking between discrete and vector worlds to incorporate raster representations, ultimately developing a virtual photogrammetry tool for mixed reality. There are many flexible deep learning utilities designed for image classification, object identification and semantic segmentation [7-12]. Of course, ML and deep learning tool aren't without flaw, but through progressive interactive training with reinforcement, transfer learning [13] and input homogenization, they can be convinced to perform well beyond expectations. Linking virtual and real worlds, however, requires testing, tweaking and reinforcement which under normal circumstance would require significant team involvement which cannot always be consumed in parallel and must be carefully managed.

Rapid prototyping through converging realities

The process of developing and proving solutions for research objectives similar to this do not need to rely on the real world initially. Through abstracting the process, there is no reason why individual components from virtual and real-world utilities or data sources can't be interchangeable. For example, the virtual photogrammetry (VP) module introduced by for the Unity game engine [14] and LiDAR scanning [15] produce data which differs only by imperfections in the that surfaces they scanned. Although virtual models do produce ideal surface maps, there is a little challenge in introducing imperfections. The premise of converging realities [6] is that tasks which are constrained by serial time, equipment allocation and team capacity in the real world aren't subject to the same constraints in the virtual world. If you have one drone in the real world, you can generate data specific to the target buildings at the rate which the camera is capable of capturing UV and point-cloud data. The properties of the building and environment are immutable, the weather and lighting are situational, and the rendering is not controllable by the pilot.

In contrast, virtual worlds are constrained only by the amount of processing time and devices that can be afforded to the project, data collection can be automatic and in parallel, and the environmental and target features are mutable. The virtual world can be spawned randomly with rendering material types, resolution and imperfections unique to a given instance; even the target can be sampled from a repository of buildings. Drones in the virtual world don't need pilots, nor do they have physical representations meaning more than one can collect data during a single collection. Unity can be compiled for Linux machines which enable next-to-nothing cost parallel processing. In short, by the time the real-world pilot has travelled to the site and generated data for a single target, the virtual world can produce thousands of fully segmented data sets from any number of targets with a flexible selection of the characteristics that are otherwise immutable in the real world. This process is by no means perfect and not implicitly transferable to the real world, but it lays the foundation of progression towards a commercially viable tool.

Image homogenization and a recurrent approach to committee ensemble creation

The aim of attempting to converge realities is to take actions which are applicable to virtual world data applicable to data from the real world by gradually blurring the lines between the two. The real and virtual world renderings are never going to be identical, and therefore, training models on purely virtual data alone would likely be unfruitful. However, between readily available rendering material images and fast style transfer (FST) CNNs, creating a progressive interactive training ensemble can be made easier. FSTs learn the common characteristics of a given training image's artistic style by comparing it with thousands of images with distinct artistic styles including works from historically famous artists. Once a model has been trained, an image can be converted from its raw state to a style-transferred equivalent (Figure 1).

The application of FSTs in this framework is not to translate images into interpretations based on famous works of art but rather to standardize the mixed-source classification and segmentation images which are used for training. By generating a collection of FSTs each classifiable object's optimal style for classification may be identified. For example, sake using (Figure 1) styles: identifying brickwork envelopes may favour Hokusai when constant air volume units on the roof may favour Picabia. In the proceeding sections, observations from Seyedzadeh et al. [17] and Progressive Interactive Training ensembles are discussed in greater detail, however, for the moment they can be considered another parallel training exercise to loosely produce classifiers for a Bayesian Committee. Committees are a novel approach to combining multiple estimates to increase estimation accuracy. However, the initial envisaged model is simpler than a true committee in the sense that it is a mixture of educated guesses via simplified classifiers recurrently delegating images to self-propagating specialists and outright recurrent brute force. A simple example of a static type of committee could be solving a linear regression problem with multiple estimators using unsupervised learning to create a clustering model for identifying homogenous datasets. For example, k-means clustering may be elbowed such that an optimal number of clusters for the training data are identified. Using the clustered data an estimator for each can be created which should have a bias towards data k-means fits within those clusters. This will not guarantee precise predictions but reduces mean squared error over all estimators.

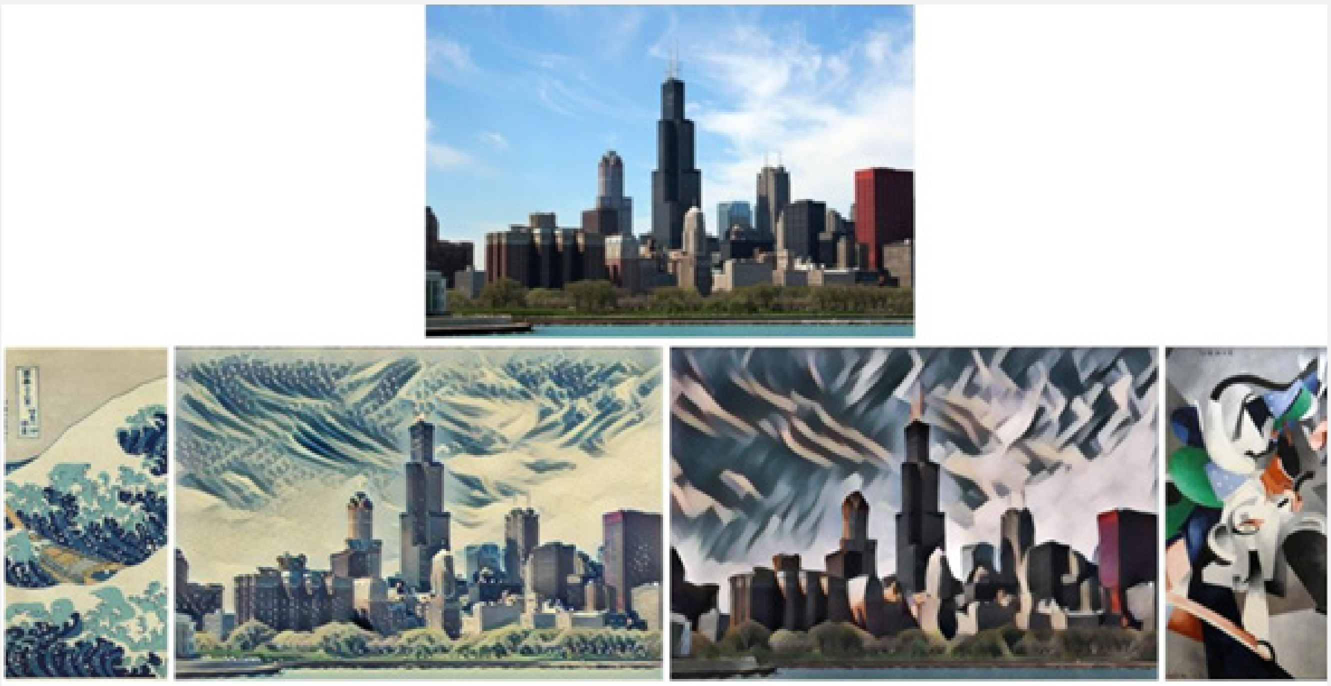

Figure 1: Lengstrom Fast style transfer examples: Left Katsushika Hokusai's The Great Wave, Right Francis Picabia's Udnie [16].

Transfer learning / reinforcement

Mentioned previously is the marginal value of human resource where opportunity costs are effectively doubled for failure. If a team member works on one project in favor of another and it is a success, there is some form of return, if they fail then ostensibly two projects have failed. This does not necessarily apply to computer processes. Putting aside perhaps being instantiated as a first-order interest, computers only have the second-order interest of keeping the user happy. Additionally, computers aren't opportunistic job candidates, they don't care what job they take regardless of process intensity or their qualification to do it. Finally, they have no external interests and as such, their work time may be perpetual. Therefore, opportunity cost is not particularly sensitive to time or capacity, only available finance. These characteristics lend themselves to recurrent and on-the-fly training methods such as reinforcement [18] and transfer learning [19], or bespoke model creation. Considering the Fast Style Transfer (FST) style committee approach mentioned previously, an FST training on a high-performance graphics processing unit should complete 1,000 iterations in 8 to 10 hours. Given the average response time for data requests in the construction industry is ten days, this time is negligible even for serial training. Seyedzadeh, et al. [3,17] researching integration of multi-objective optimization with gradient boosting regressors for energy performance analysis considered a pessimistic reinforcement learning model. They observed potential for a 500% improvement in mean square error using a significantly smaller training data set by generating target-specific emulators. This did require generating real model outputs for a sample of scenarios for each target however, the time is not significant when considering the high costs associated with retrofit decision-making. This does not fall directly under the category of reinforcement, but the principle of clustering discussed previously and opportunity for semi-supervised model aptness assessment lend themselves to the reinforcement principle. The takeaway from here is model accuracy improved to meet the needs of the project without attempting to produce a universal model.

Progressive interactive training hierarchal ensemble

Progressive interactive training (PIT) is a model for managing ensembles such that only the components which are not performing as expected are refined as new data becomes available. For example, if the CAV identifier is performing well but the envelope identifier is not then only the envelope identifier need be retrained. A more concrete example of this might be a network for identifying envelope renderings. If a roughcast identifier is working well but not a brickwork model, then retrain the brickwork identifier. In broader terms, the proposed classification model is a hierarchal and recurrent PIT. Before attempting to classify specific materials, it would make sense to first attempt to classify different characteristics such as whether an envelope is opaque or transparent. Once this has been identified the ensemble can delegate prediction to a specialized identifier. In the case of glazing, this may be the general identifier of the glazing type or perhaps exclusively a frame material identifier. In the case of the latter, the ensemble may then delegate to specialists for the identified material type. This is where reinforcement can be useful for recurrently training parent models. Once a specialist has identified a certain feature it can feed the data back to parent for reinforcement. Transfer learning for this framework is a solution to blurring the lines between the real and virtual datasets. Primarily used with targets with existing BIM models and capacity for data collection, the existence of elements can be inferred reasonably by localizing the camera in the virtual model relative to the real-world camera and comparing spatial data for known elements in the virtual environment. For example, if a CAV system is known to be present on the roof of a building from the virtual model and it is inferred from the comparing virtual and real-world spatial data that it is likely the CAV is present on the roof then the segmentation or classification models' prediction can be supervised with reasonable confidence. If the CAV is not identified, the image and samples of images targeting the same object can be used to reinforce the model. If the CAV is identified, then the image can be tagged as correct and fed back to the trainer. This can reduce noise in the training data without introducing bias. Amongst normal benefits expected form reinforcement and transfer learning, this may reduce the hierarchal ensemble model or alternatively indicate that a sub-network of specialists is required for identifying certain objects. For example, CAV model-specific classifiers.

In summary. The model suggested for this framework is a hierarchal progressive interactive training ensemble. Ostensibly a simple form of Bayesian network relying on characteristic-specific classifiers to delegate images to specialists who can attempt to classify objects or pass them further down the hierarchy. This process is reinforced by producing more biased "specialist" classifiers which upon correctly classifying object propagate back up the hierarchy to reduce noise from training datasets and simplify the ensemble. Progressive interactive training reduces the number of training exercises required and the positions in the hierarchy that require retraining.

Virtual photogrammetry

The Image Processing library from Pour Rahimian et al. [5] contains the virtual photogrammetry component. It is a game engine embedded LiDAR type scanner emulator. For a given camera it uses ray casting to iteratively produce a point-cloud as the camera moves through the environment. Each point in the cloud can be linked via StrathIFC IFC to Unity linking library to its constituent IFC object, game engine object, relative Cartesian and Barycentric coordinate systems and pixel in any given image the point is associated with. In line with the Unity game engine's discrete geometry system, the library supports scalable voxel reconstruction of the point-cloud with primitive objects. The latter is was initially intended as a novel feature for training a recurrent reconstruction neural network though not envisaged to be used in this project. Its secondary function is to provide an interface for splicing the information layers to create composite 2D images and voxel models of the generated data (Figure 2).

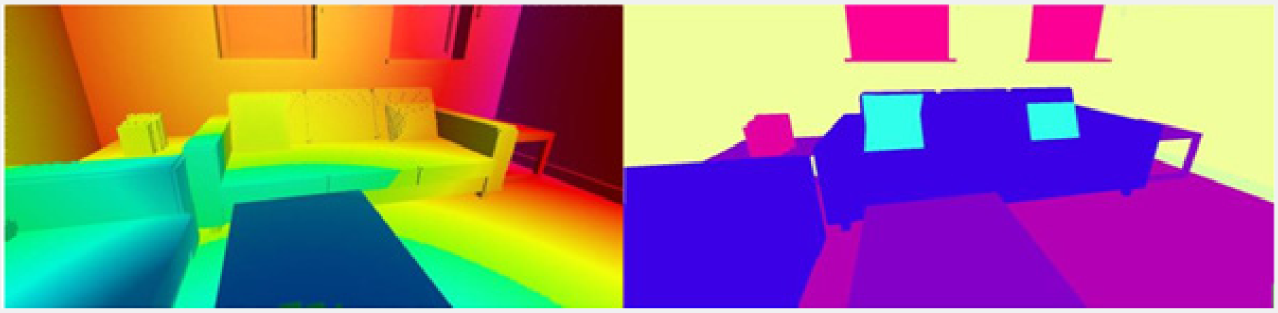

Figure 2: Voxel reconstruction with heat map of Unity-IFC data. IFC classified 2D composite image.

Summary

The proposed approach for as-built BIM model is to create a hierarchal progressive interactive training ensemble for object identification and semantic segmentation. To reduce risk and expedite the development process, the proposal incorporates a concept informally labelled as converging realities. The process is an attempt to gradually blur the lines between virtual and real worlds such that time- intensive and costly real-world processes can be delegated to virtual worlds. Reinforcement and transfer learning combined with a fusion of existing BIM models will expedite and facilitate gradual integration of the real world to the learning process. Parallel computing processing is to be used such that opportunity cost need only be measured meaningfully in money, removing expertise, capacity or time.

Acknowledgement

None.

Conflict of Interest

No conflict of interest.

References

1. Curry E, Donnell J O, Corry E, Hasan S, Keane M, et al. (2013) Linking building data in the cloud: Integrating cross-domain building data using linked data. Adv Eng Informatics 27(2): 206-219.
2. Seyedzadeh S, Rahimian F P, Glesk I, Roper M (2018) Machine learning for estimation of building energy consumption and performance: a review. Vis Eng 6: 5.
3. Seyedzadeh S, Rastogi P, Pour Rahimian F, Oliver S, et al. (2019) Multi-Objective Optimisation for Tuning Building Heating and Cooling Loads Forecasting Models, in: 36th CIB W78 2019 Conf. Newcastle.
4. Oliver S, Seyedzadeh S, Pour Rahimian F (2019) Using real occupancy in retrofit decision-making: reducing the performance gap in low utilisation higher education buildings, in: 36th CIB W78 2019 Conf.
5. Pour F, Rahimian, Chavdarova V, Oliver S, Chamo F (2019) OpenBIM-Tango integrated virtual showroom for offsite manufactured production of self-build housing. Autom Constr 102: 1-16.
6. Pour F, Rahimian, Seyedzadeh S, Oliver S, Rodriguez S, et al. (2020) On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning. Autom Constr 110: 103012.
7. Hazirbas C, Ma L, Domokos C, Cremers D (2017) FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture, in: Lect. Notes Comput. Sci. (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics): 213-228.

Citation: Stephen Oliver, Saleh Seyedzadeh, Farzad Pour Rahimian, Nashwan Dawood, Sergio Rodriguez. Cost-Effective as-Built BIM Modelling Using 3D Point- Clouds and Photogrammetry. Cur Trends Civil & Struct Eng. 4(5): 2020. CTCSE.MS.ID.000599.
DOI: 10.33552/CTCSE.2020.04.000599.

8. McCormac J, Handa A, Leutenegger S, Davison A J(2017) SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation. In: Proc IEEE Int Conf Comput Vis: 2697-2706.

9. Ros G, Sellart L, Materzynska J, Vazquez D, Lopez AM (2016) The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In: Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit: 3234-3243.

10. Gupta S, Girshick R, Arbelaez P, Malik J (2014) Learning rich features from RGB-D images for object detection and segmentation supplementary material, in: Comput. Vis-Eccv: 345-360.

11. Soltani MM, Zhu Z, Hammad A (2016) Automated annotation for visual recognition of construction resources using synthetic images. Autom Constr 62: 14–23.

12. Hongjo K, Hyoungkwan K (2018) 3D reconstruction of a concrete mixer truck for training object detectors. Autom Constr 88: 23-30.

13. Sundaravelpandian S, Johan S, Philipp G (2018) Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction. Adv Eng Informatics 38: 81-90.

14. Unity Technologies, Unity (2018).

15. Wang J, Sun W, Shou W, Wang, Wu C, et al. (2015) Integrating BIM and LiDAR for Real-Time Construction Quality Control, J. Intell. Robot. Syst. Theory Appl 79: 417-432.

16. Engstrom L (2016) Fast Style Transfer in TensorFlow.

17. Seyedzadeh S, Pour Rahimian F, Rastogi P, Glesk I (2019) Tuning machine learning models for prediction of building energy loads, Sustain. Cities Soc 47: 101484.

18. Sewak M, Sewak M (2019) Introduction to Reinforcement Learning, MIT press Cambridge, UK.

19. Pan SJ (2014) Transfer learning, in: Data Classify Algorithms. Appl IGI Global 537-570.