S. Chien, S. Choo, M. A. Schnabel, W. Nakapan, M. J. Kim, S. Roudavski (eds.), *Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA 2016,* 445–454.

EDITION-ORIENTED 3D MODEL REBUILT FROM PHOTOGRAPHY

Giving affordance to 3D capture

JOAQUIM SILVESTRE, FRANÇOIS GUÉNA and YASUSHI IKEDA
Keio University, Fujisawa, Japan
j.silvestre82@gmail.com, francois.guena@maacc.archi.fr, yasushi@sfc.keio.ac.jp

Abstract. The topic of this paper is about a technique to turn pictures into an intuitively modifiable 3D model. The research employs an analytical method using algorithms to conceptualise and digitalise architectural spaces in order to highlight parametric shapes. Usually, from one group of digital photos, photogrammetry techniques produce a 3D-model mesh through a high-density 3D point cloud. This discordance between our intuitive partitioning of the mesh and its bare polygonal structure makes it interact poorly compared to the affordance of shape and component in our daily experience. Through a capture device, a visualisation of architecture in a digital data form is produced. They are processed by computer vision algorithms and machine learning systems in order to be refined into a parametric model. Parametric elements can be described as a compound of formulas and parameters. By keeping the formula and changing the parameters, these elements can be easily modified in a range of likenesses. After being detected during scans, these shapes can be adapted to fit the intention of the designer during the design phase.

Keywords. Photogrammetry; convolutional neural network; 3D model; design tool.

1. Introduction

Local surroundings in architectural design are often recorded by building surveyors prior to design sessions. Based on these records, and visits to the site, architects start to imagine possibilities for the future. Records are abstractions of the real site, so they need to be reconstructed in the mind of

the designer. With a complex site, it can be hard to work out how to develop it. A good way to help the visualisation process can be 3D reconstruction based on photography.

Usually, architecture practice is about creating new building. It implies that existing old architecture around is demolished. This policy of Tabula Rasa is exactly the opposite of a continuous design that: "generate in the flow of what is already there […]" (Wood, 2007). In the framework of the Algorithmic design, this architecture practice can be considered as minimal intervention in a system to solve a problem instead of fully create a costly solution from scratch. In order to move toward this idealistic practice, there needs to be a more comprehensive and integrated access to site data. This could be the key for an Algorithmic design sites intervention which carries and transfers weight to the existing constructive elements instead of wasting energy to move them away.

Photogrammetry is a technology that is becoming more accessible. However, despite the fact that generated 3D models can be very accurate, their internal structures are dissonant with our intuitive perception of spaces and objects. In the end, it makes 3D shapes that are hardly editable for architectural intervention. For instance, if you want to change the radius of what you see as a tubular section, it's often necessary to remodel the site with a built-in CAD software command upon the 3D scan. For a CAD user, it means the ability to modify parameters of shape directly. Without any remodelling or fuzzy vertice selection process, he can extrude a table or remove a window.

The 3D scan ease the precise and complete capture of geometric data but misses the semantic and constructive aspects. Previous research has brought innovative solutions to cope with this problem. Without being exhaustive: Tang et al (2010), Shen et al (2012) and Andelo Martinovic (2015) present similar goal to this research. This work can be related to "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques". This review presents the overall context and the challenges of 3D reconstruction. This paper will explore a slightly different approach since it considers pictures as raw material and convolutional network as a technique to segment and identify the architectural component in the surroundings. Concerning the strategy, it may be similar to "Structure Recovery by Part assembly" except that it copes with the problem of detecting scattered parts of the component in various pictures instead of one.

Finally, these methods are slightly different. They use 3D point cloud analysis, computer vision oriented and shape grammar extraction but none of them use Deep Convolutional Neural Network (ConvNet) (Nielsen, 2015). The experiments conducted in this paper will focus on architecture models of

small volumes and enclosed spaces. Real scale can only be processed with large training data that is costly. So the training database is reduced to a number of easily testable cases. The results obtained using the limited database constraints can serve to speculate further scaling up to a larger system.

2. Overall process

This process, from data acquisition to 3D reconstruction, borrows Photogrammetry technique and links it to ConvNet. To create ConvNet, it is necessary to have a ConvNet architecture to train using a data set. As this uses ConvNet in two steps of the process, specific sections are focused on details of data set content and ConvNet architecture.
Like photogrammetry, the input data is a set of pictures that overlap each other. Homologue points between pictures are calculated. Then with internal and external calibration, the relation between the cameras is retrieved. Our process uses the same initial steps. It allows us to get pictures organised by their proximity to each other within the captured object space (see Figure 1). After this step, the photogrammetry process continues with a dense correlation in order to generate dense point cloud. Points of the cloud are connected through a mesh triangulation algorithm. These steps create the effective 3D mesh. The method described in this paper replaces the dense 3D point cloud step and all the following steps.

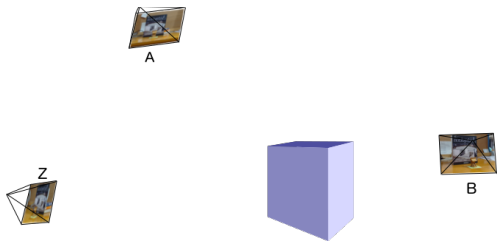

Figure 1. Bundle Adjustment of the camera position.

The images embedding their camera coordinate will be segmented into crossed categories based on the likeness, same objects amongst different images belong to the same segment, and type, same type of object in one picture belong to the same meta-category. Segmentation will be done by pixel labelling with ConvNet like in the work of Farabert (2013).

ConvNet requires data in a uniform format. The 3D coordinates of the camera position can't be provided next to a picture images. Even if they can be all converted to tensor matrix format, neural network components are not made to process 3D coordinates vector data along with a matrix of color data of each pixel. From a recent paper of John Flynn (2015), a way to implicitly

embedded camera coordinates and orientation within the pictures' image is discussed. This technique is used for the same need of ensuring data uniformity in their paper. Like in figure 2, it applies a perspective deformation on a picture (B) in order to match with the point of view of the previous picture (A). In other words, image (B) is reprojected into the target of the camera's picture (A). This system implicitly provides the relative coordinates of the picture (B) according to the picture's (A) coordinates into the picture (B). Since the coordinates of the cameras' pictures are provided with internal and external calibration, pictures can be chained based on their geographic proximity. Each picture will be re-projected in the subsequent picture space in the series as long as there is data available.

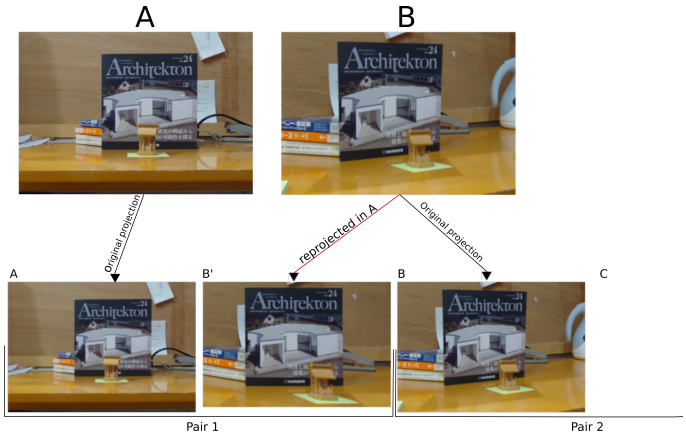

Figure 2. Reprojection chain.

The first ConvNet, (Twin consensus segmentation network[1]) will receive pairs of pictures: one in original space, another reprojected. It's segmentation ConvNet with an interaction of weight sharing at the full connection layer. The weight sharing will harmonise the segmentations between the two points of view in order to propagate the same segmentation class between different views of the same object while it differentiates objects that belong to the same class according to their relative position to each other. Indeed, their geometric deformation will extend their segmentation group but the camera movement will separate the object through their relative movement to each other. For instance, object next to each other and belonging to the same segmentation class are dissociated in the output as two segments of two different objects. The output classes of segmentation are divided in 8 common architectural components: walls, roofs, doors, windows, stairs, handrails, post, beams and 13 special temporary object classes. This excludes the ground, sky and "outside" classes that are potentially infinite seg-

ments. These three classes are not producing 3D model. Each of the segmentation classes are extending their area from picture to picture until the features appear in a discontinuous pictures area or no features of this class appear anymore. When features of this class reappear in another picture or in a discontinuous space, a new picture element's object repository is open. The special object classes will be temporary classes that stock simultaneously aggregated super pixels in object's facets per camera into different unknown objects with different label.

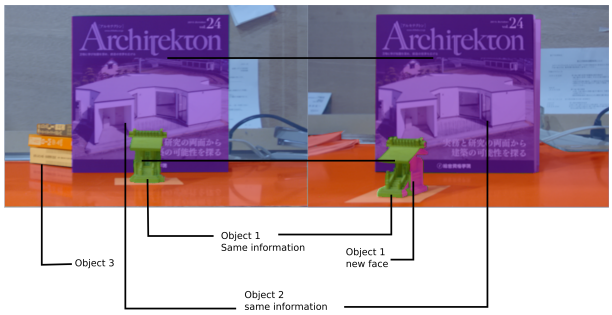

Figure 3. Segmentation of 2 adjacent pictures.

The output segments of the ConvNet are remapped in their original picture space and bind with corresponding camera coordinates. Then the picture elements of each object are stored in a database that keeps image segments of each continuous along adjacent picture segments. Each picture elements group are stored under their class label and a component identification number. Under each picture element of a group, the camera coordinates and the corresponding depth-map are retrieved from the initial input data.

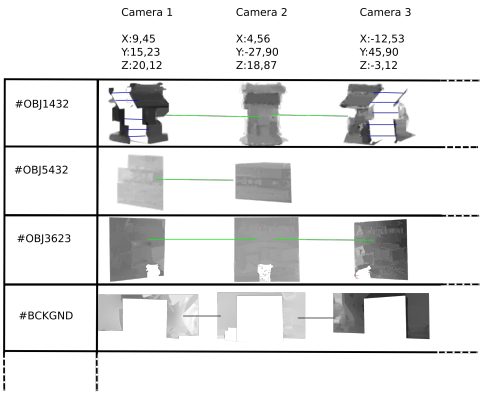

Figure 4. Storage of the Object Pictures Elements: green line link different point of view of the same "face", purple lines indicate seam of part, black connect background.

To retrieve the depth-map, we could use OpenCV command library on two close pictures, or photogrammetry software to reconstruct a temporary mesh from which a virtual camera extracts a depth-map on the same coordinates of the taken pictures. But in order to focus on the most ambitious step of the process and to avoid a technical convoluted process, we chose to use a plenoptic camera (Lytro Illum) that provides directly the depth-map matching to the color pixel of the pictures taken. Inherently, this camera produces already low level three dimensional information (depth-map) that ease and shortens the whole process described here. In future works, we aim to work with various capture systems like a 360 or stereorcopic camera.

Then, the second ConvNet (Depth feature detector[1]) is used to match depth features of primitive shape inside the unfolded puzzle of picture elements of each object. The features are tracked along the chain of picture elements in order to retrieve the closest primitive shape that can create a similar set of features. Since picture elements will not cover all the facets of the object, the ConvNet set a probability ranking of possible shapes according to the group of features detected. Then, amongst the possible shape that constitutes the object, the reconstruction algorithm selects shapes that are not overlapping each other. According to the most plausible ingredients, the algorithm will propose a recipe for reconstruction. This part is not yet implemented for complex shapes. For now the detection of shape is quite limited so the reconstruction is simple. It starts from the ground then block, tube, cone and sphere shapes are stacks one upon the other.

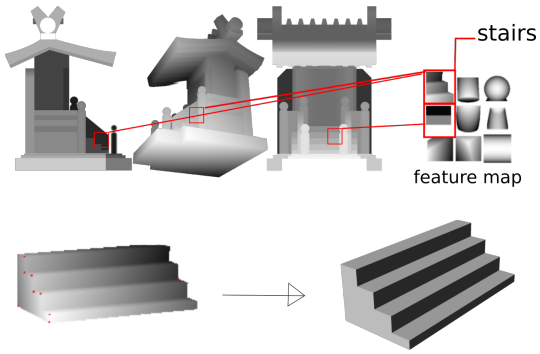

Figure 5. Details from the different views trigger features in the ConvNet layer. Once the type of shape is detected, the appropriate computer vision algorithm extracts the key point for reconstruction.

The reconstruction strategy depends on the CAD software used for reconstruction. It maps features of the depth-map with a corresponding command in the software and retrieves the coordinates of useful 3D points required by

the command. For instance, in Rhino 3D modeling software, a parallelepiped is defined with at least 3 points. The second ConvNet detects the shape and the area in the blue print that belongs to this shape. In this area, the signature of a corner is searched. The coordinate of the corner in the depth-map space is defined in 2.5D: x,y and the z position according to the depth-map. The identification of the pictures chunk from where the signature comes from corresponds to a camera position. With the 3 corner points and the 2 or 3 corresponding camera, the points are re-projected in the 3D space through the camera position. For now, the program is limited to detect and retrieve key construction points and lines of primitive shape: cones, spheres and parallepipedic shapes. Theses shapes are easy to edit and present in all CAD software. The step that connects the output of the system with API of CAD software is not implemented. Actual output is just shape names and a series of coordinates in a text file.

3. Data set construction

Data set construction is a difficult point in machine learning. To cover lots of cases, it needs to include a large number of pictures. In order to keep this project to feasible proportion, the detection expectation has been kept low. For the Twin consensual segmentation ConvNet, only a very limited number of examples of each class have been kept. Other papers that propose solutions for segmentation show good result with 400 pictures and corresponding ground truth segmentation.

For Depth features Detector, the training set depends of the number and the type of command from CAD software that are allowed to be used to reconstruct the captured architecture.

3.1. TWIN CONSENSUS SEGMENTATION DATASET

Images in the data set are a pair of close views with their corresponding segmentation. Segmentation is partially done with GrabCut[2] for the foreground objects and SLIC[3] for background partial elements. The 13 special temporary object class are attributed in depth order in the first picture of the chain − the closest object from the first camera in the chain will be the object #1. Pictures are mainly interior and exterior pictures of architecture building. No humans or animals are present in the pictures. Mobile elements are captured in only one state and not displaced between shots in a chain. Pictures chain are varying between 2 and 10 pictures. The overlap of pictures is similar to photogrammetry requirements: 60% of side overlap with 80% of forward overlap.

Meanwhile, the dataset is a decent size for training (600 images at least) and testing (200). A smaller temporary dataset of 40 pictures with just ground, background wall and foreground objects categories have been made just to test the consensus segmentation between pictures of a minimalist architectural model.

3.2. DEPTH FEATURES DETECTOR DATASET

Images in the dataset are depth-map pictures of primitive shapes: cones, cylinders, spheres and parallelepipeds of various sizes with various positions and angles of view. Labels are given by the name of the command used in the CAD software to produce the shape. Here, the goal is to produce a good feature description of a depth-map from a primitive shape. The picture format is 500*500 with levels of grey and an alpha channel. The depth-map is generated with a Rhino Python script that runs render of primitive shapes with random parameters and random points of view on the Depth channel. The ease to produce these images allows us to generate 10,000 of them.

4. Network architecture

ConvNet architecture is composed of various layers. Each layer has a specific function in the task. Usually, higher layers detect elementary patterns and lower ones associate these patterns into a bigger one. During the training phase, the ConvNet creates its own classification. By “own” we have to understand that it doesn’t necessarily follow a human classification logic. If a Deep neural network had to do the species classification, it will probably base is classification on other characteristics than taxonomy scientist did. It would result in organisation of the class slightly different from mammals, oviparous and vertebrate. It produces an efficient classification rather than an intellectually satisfying one.

Another important element is the convolution layers. These produce feature maps. These maps are abstract representations of patterns that are shared amongst images. We could think of features as an informative detail of the shape but they are hardly representable since they are an abstract relationship between pixels.

4.1. TWIN CONSENSUS SEGMENTATION NETWORK

Without detailing the function of each layer, the most important layer in the network is the Spatial pyramid. It changes the resolution of the input images and processes these different resolutions through a cloned network with shared weight. It allows us to create features of bigger scale: features that express the general image organisation. In the segmentation task, it avoids

incongruous errors like detecting people in the sky. Another critical point is the interaction between the twin networks. Objects in the foreground will tend to move more between two pictures. Disparities in the global area of the spatial pyramid between parallel layer will tend to detect the closest object to the camera.

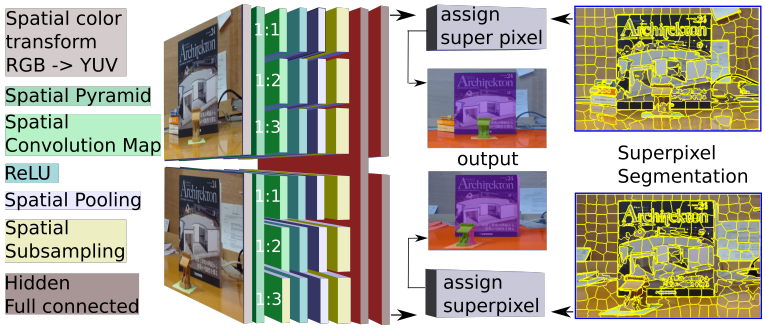

Figure 6. Architecture of the Twin consensus segmentation network

The architecture is here more conventional. Convolution layers followed by pooling ones is a very common way to extract the hierarchy of features. The input image is a depth-map and the output for classification is the name of the different primitive shape.

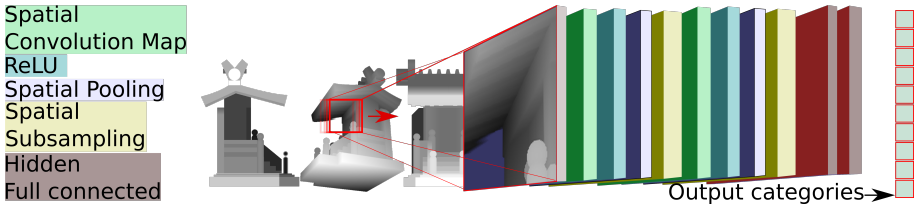

Figure 7. Architecture of the Depth Feature Detector Deep Neural Network (ConvNet)

5. Conclusion

To conclude, we can't, unfortunately, says that this system works. Any shape elements haven't yet been detected. It is likely due to the numerous glitches in the depth-map produced by the plenoptic camera (Lytro Illum). Some attempts has been made with a projected light system, but the same problems of depth-map quality have been encountered. Some parts of the depth-map are missing and the resolutions are even lower. One solution could be to use LiDar (light detection and ranging) but they are costly and it doesn't fit with the original idea of a low-cost capture system. As said previously, another cost-effective solution could be to generate the depth-map through a photo-

grammetry software. It's a complicated and inelegant solution but it is worth being tried since the attempt to ease the technical process was the source of failure.

Coherence between a 3D model reconstruction structure and an intuitive understanding of architecture could drastically extend the ability of Algorithmic architects. As a matter of fact, the design process alternates between the conceptualisation and the visualisation phases. The Computational Thinking[4] approach of design involves a "computational conceptualisation" which fits to envision architecture as structured data. Therefore, our perception and ability to act on the environment are deeply bonded, an algorithmic control of architectural data will stimulate more involvement of computational thinking into the phases of design. In the end, it leads to an Algorithmic design practice: an automated manipulation of data in order to solve a design problem. More practically, intervention on site specific problems like building modifications or extensions become easier. For instance, an office layout planer can reorganize the existing furniture and working space of a level. The real world becomes a repository of 3D models that are just waiting to be scanned.

Endnotes

1. This name is not related to any papers. It's only made up name to ease the description.
2. Interactive foreground extraction algorithm using iterated graph cuts.
3. Superpixel Linear Spectral Clustering, segment image into superpixel according to perceptual meaningfulness
4. Method to generalise solution of problems in the intent to solve them with computer.

Acknowledgements

I would thank the Taikichiro Mori Memorial Research Fund to provide us the fund to buy a plenoptic camera.

References

Flynn, J.; Neulander, I.; Philbin,J.; Snavely, N.:2015, DeepStereo: Learning to Predict New views from the World's Imagery, *arXiv*.

Farabet,C.; Courpie, C.; Najman, L.; LeCun, Y.: 2013, Learning Hierarchical Features for Scene Labeling, *IEEE Trans*.

Martinović,A.: 2015 Inverse Procedural Modeling of Buildings (Thesis), Arenberg doctoral school.

Nielsen, N. M.: 2015, *Neural Networks and Deep Learning*, Detremination Press.

Shen, C-H.; Fu, H.; Chen, K.; Hu, S-M.: 2012, Structure recovery by part assembly, *Proceedings of ACM SIGGRAPH Asia* 2012

Tang, P.; Huber, D.; Akinci, B.; Lipman, R.; Lytle, A.: 2010, Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review techniques, *Automation in construction*.

Wood, J.: 2007, *Design for Micro-Utopias: Making the Unthinkable Possible*, Gower.