Title	Automatic generation method for processing plants
Author(s)	Nomura, Kentaro; Miyata, Kazunori
Citation	Proc. of IEVC2010, 3B-6: 1-6
Issue Date	2010-03-07
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/12074
Rights	Kentaro Nomura, Kazunori Miyata, Proc. of IEVC2010, 3B-6, 2010, pp.1-6. 本著作物は画像電子学会の許可のもとに掲載するものです。
Description	

AUTOMATIC GENERATION METHOD FOR PROCESSING PLANTS

Kentaro NOMURA[1)]*Kazunori MIYATA*[2)]

[1)] School of Knowledge Science, Japan Advanced Institute of Science and Technology
[2)] Center for Knowledge Science, Japan Advanced Institute of Science and Technology

ABSTRACT

A processing plant consists of massive parts including tanks, pipelines, processing columns, frames, and so on. This paper reports a method for automatically generating the landscape of a processing plant from a 2D sketch input and some control parameters. This is difficult to implement with conventional procedural methods. The results show that the landscapes of a processing plant are satisfactorily represented, while some detailed parts, such as valves, steps, and branching pipelines, are not generated. The generated 3D geometry data are useful for constructing background scenes in movies and video games, and are also applicable for pre-visualizing a landscape to construct a processing plant.

1. INTRODUCTION

Recently, massive geometric models are required to produce high-definition computer generated imagery. This paper reports a method for automatically generating the landscape of a processing plant from a 2D sketch input, which is difficult to implement with conventional procedural methods.

The generated 3D geometry data are useful for constructing a background scene in movies and video games, and are also applicable for pre-visualizing a landscape to construct a process plant.

2. RELATED WORKS

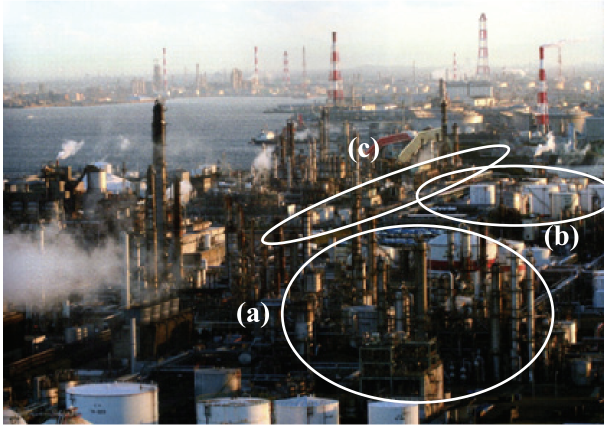
Procedural Inc. sells the software, *CityEngine* [1], for generating a large scale 3D urban environment procedurally, including street networks and 3D buildings. Frischer et al. have applied this software to rebuild highly detailed ancient Rome at the peak of the Roman Empire [2].

Parish and Müller proposed a system using a procedural approach based on L-system to model virtual cities [3]. Müller et al. reported a novel shape grammar for the procedural modeling of building shells to obtain large scale city models [4]. They also introduced algorithms to automatically derive 3D models of high visual quality from single facade images of arbitrary resolutions [5].

Most of these procedural methods generate virtual cites from two-dimensional images. In contrast, this paper proposes a method to generate a processing plant which consists of tanks and pipes from two-dimensional sketch input.

3. GENERATION METHOD FOR PROCESSING PLANT

A processing plant consists of massive and interlacing parts including tanks, pipelines, processing columns, frames, and so on as shown in Fig.1.



#1: (a) Processing Unit, (b) Tank Yard, (c) Pipelines

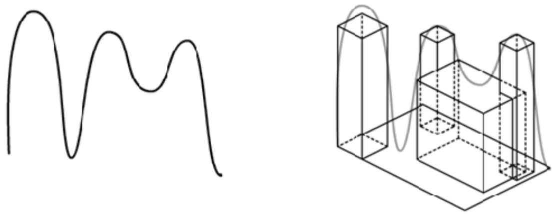
#2: a) Tower, (b) Tank, (c) Pipeline, (d) Frame

Figure 1: Example of Processing Plant (Ref [6])

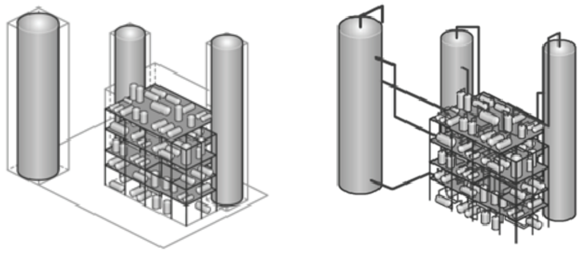
This method automatically generates 3D geometric data for each component, from a hand-drawn 2D sketch and some control parameters.

3.1. System Overview

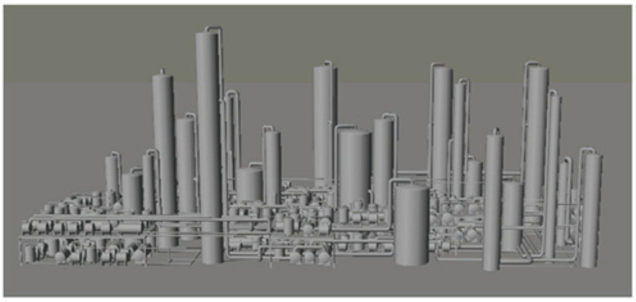
Fig.2 shows an overview of the process. 3D bounding boxes are generated from a 2D sketch. Here, a 2D sketch specifies a profile of a process plant. Next, frames are generated from the bounding boxes, then tanks are arranged on each floor. Finally, each tank is connected to adjoining tanks by pipelines.



(a) input 2D sketch (b) bounding boxes



(c) tank arrangement (d) piping



(e) example

Figure 2: Procedure Overview

3.2. Analysis of Input 2D Sketch

3D bounding boxes of the process plant are generated from a 2D sketch which is drawn with a single stroke.

First, the input 2D sketch is partitioned into columns as shown in Fig.3. The specified 2D sketch is scanned from left to right to calculate the gradient of the profile, dy/dx. A column is formed when the absolute value of the gradient is over the pre-defined threshold parameter. The steeper the gradient, the narrower the width. Here, the minimum column width is also pre-defined. We have set the threshold parameter to 0.9, and the minimum width to 15, respectively.

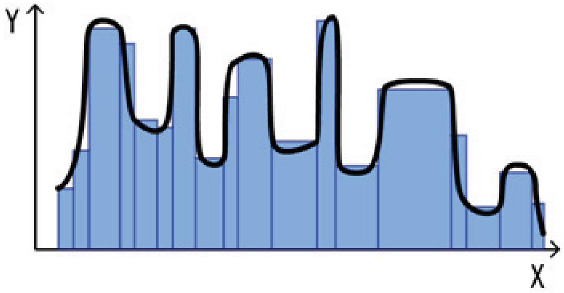
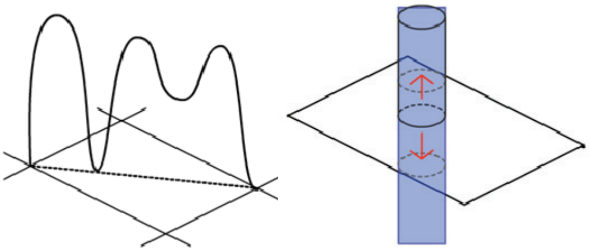


Figure 3: Partitioning of 2D Sketch

Then, the extent rectangle of the site for a plant is determined by placing the 2D sketch so as to fit the diagonal of the rectangle as shown in Fig.4(a). Here, the three dimensional viewing angle is specified by a user. Next, a bounding box for each column is placed randomly within the obtained extent rectangle of the site, by sliding its position according to the viewing direction as shown in Fig.4(b).



(a) determining the extent rectangle of the site (b) column placement

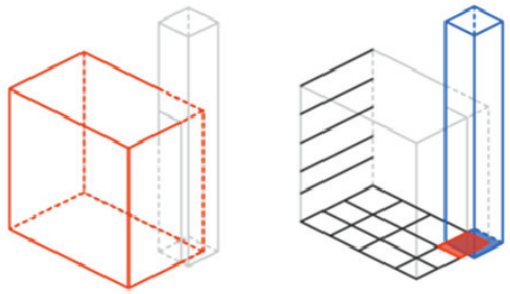
Figure 4: Partitioning of 2D Sketch

3.3. Frame Generation

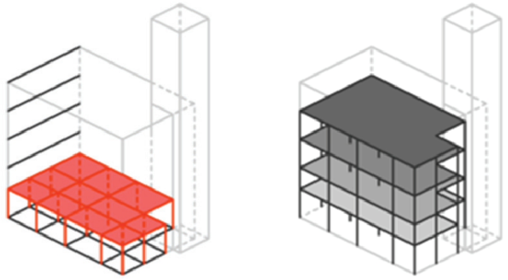
A histogram of heights is calculated from the partitioned 2D sketch. Then a threshold value of 30% is used to distinguish between columns (top 30% boxes) and frames (other boxes).

Fig.5 illustrates the procedure for generating a frame from a bounding box. A bounding box, which is labeled as a frame (red box in Fig.5 (a)), is partitioned into lattices avoiding other boxes, as shown in Fig.5 (b). Then, each floor is generated repeatedly, as shown in Fig.5 (c).

A bounding box which is labeled as a processing column is replaced with a column in the following procedure.



(a) bounding box (b) partitioning



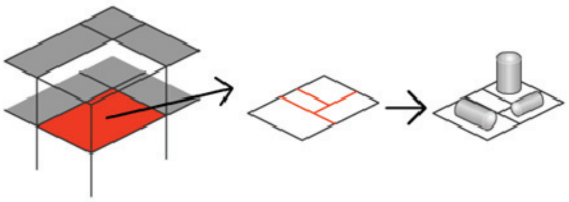
(c) floor generation (d) result

Figure5: Frame Generation

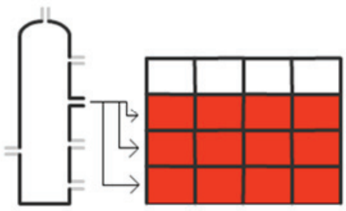
3.4. Tank Arrangement and Piping

After frame generation, each floor is recursively tessellated into smaller rectangular cells until the size reaches a threshold value, and then each cell is filled with a tank, as shown in Fig.6 (a). Each tank is randomly placed horizontally or vertically. Finally, each outlet of a tank is connected to one selected adjoining lower tank by a pipeline, as shown in Fig.6 (b).

Some control parameters, such as the number of tessellations, the occupation ratio of the tank, the tank size, and the number of pipeline connections are specified by a user.



(a) tank arrangement



(b) piping

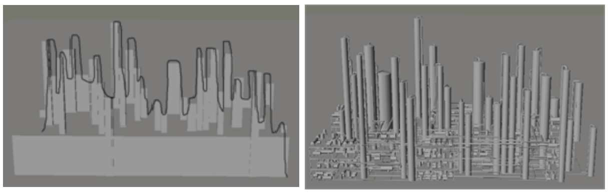
Figure6: Tank Arrangement and Piping

4. RESULTS

Fig.7 illustrates the comparison of a real sample and a generated image. The black curve in Figure 7(b) is the sketch input which is specified by tracing the real sample, Figure 7(a). It is impossible to specify the precise depth data for each processing tower in our system, but we can observe similarity between the real sample and the obtained image.

Fig.8 shows the results generated by this method. Each image is rendered using skylight illuminance. Fig.9 shows the variations controlled by some parameters, such as the number of tessellations, the tank size, and the proportions of tanks.

(a) real sample (b) traced 2D sketch input



(c) bounding box (d) generated model

Figure 7: Comparison of Real Sample and Generated Model

The processing time depends linearly on the number of pipelines, as shown in Fig.10.

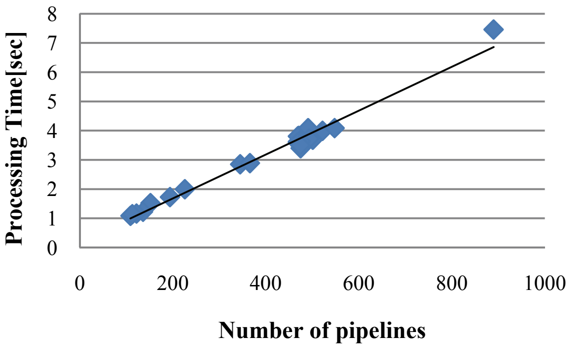


Figure 10 Processing Time

5. CONCLUSION

The examples show that the landscapes of a processing plant are satisfactorily represented, while some detailed parts, such as valves, steps, and branching pipelines, are not generated. The method does not consider collisions of pipelines and other objects, either. Therefore, some pipelines penetrate tanks and processing columns. We plan to address these issues in future research.

REFERENCES

[1] Procedural Inc., “CityEngine,” http://www.procedural.com/

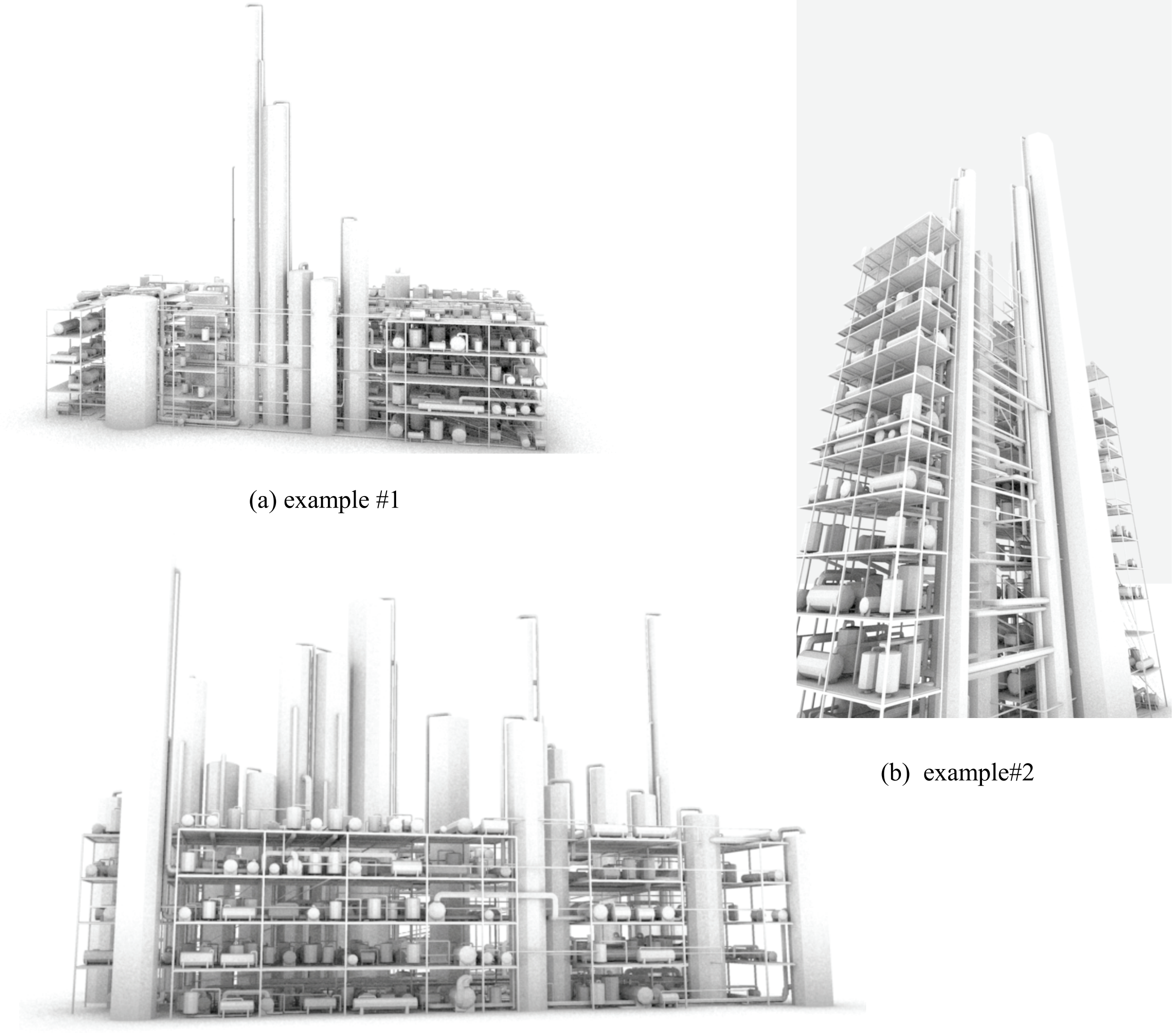
[2] Frischer, B. et al., “Rome Reborn,” SIGGRAPH2008 New Tech Demos, 2008.

[3] Parish, Y. I. H., and Müller, P., “Procedural modeling of cities,” Proc. of ACM SIGGRAPH 2001, pp.301-308. 2001.

[4] Müller, P., Wonka, P., Haegler, S., Ulmer, A., and Van Gool., L., “Procedural Modeling of Buildings,” Proc. of ACM SIGGRAPH 2006/ACM TOG, vol. 25, pp.614-623, 2006.

[5] Müller, P., Zeng,G., Wonka, P., and Gool, L.V, “Image-based Procedural Modeling of Facades,” Proc. of ACM SIGGRAPH 2007, #85, 2007

[6] Ohyama, K., and Ishii, T., “Kojo Moe,” Tokyo-Shoseki Co.Ltd, 2007 (in Japanese)

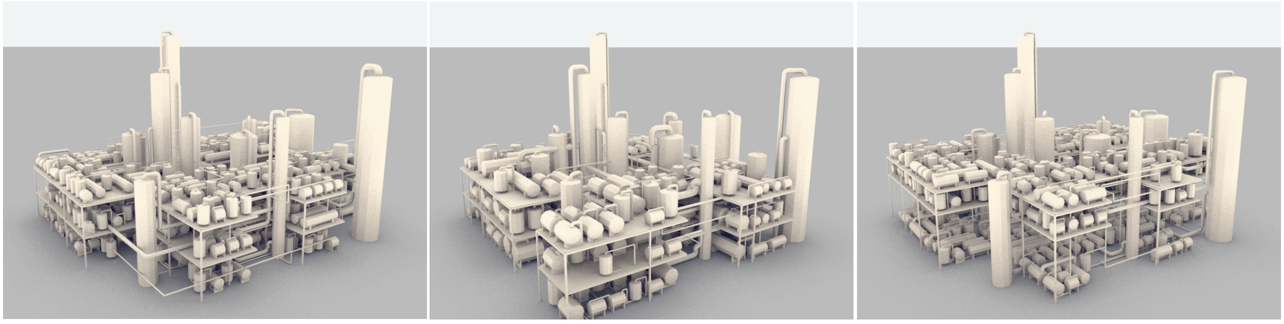


(a) example #1

(b) example#2

(c) example#3

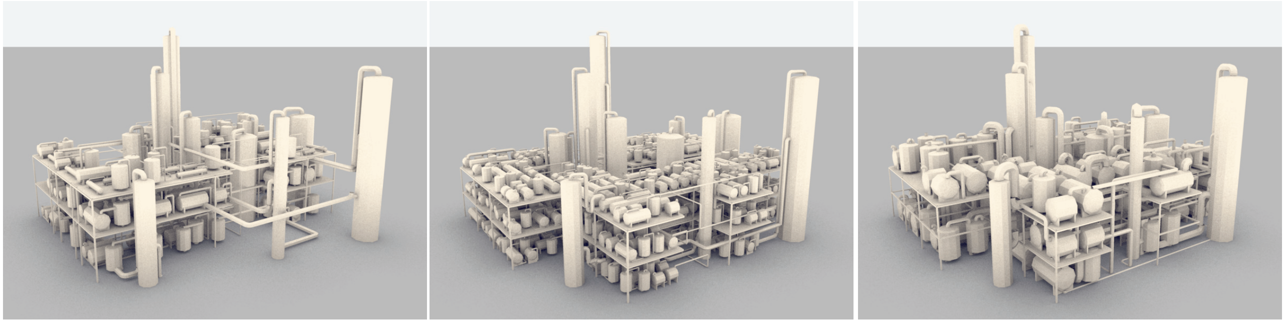
Figure8: Examples



(a) uniform frame distance

(b) varied horizontal frame distance

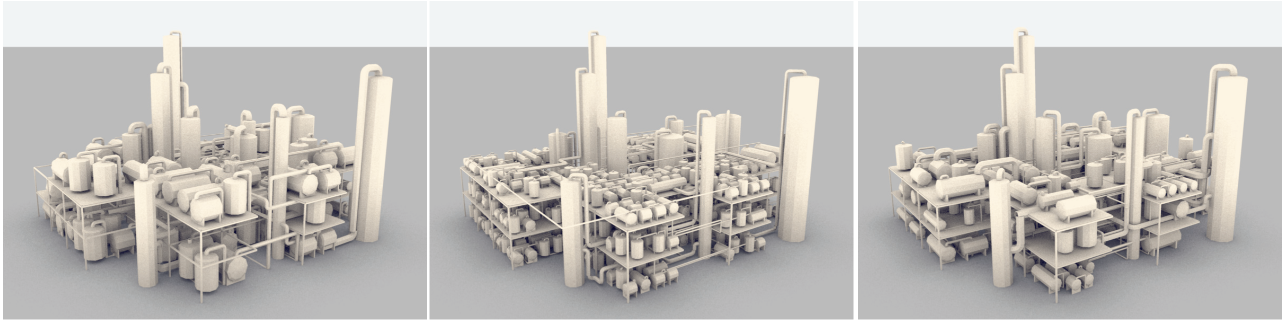
(c) varied vertical frame distance



(d) varied horizontal and
vertical frame distance

(e) fine cell tessellation
(minimum cell size = 0.8m)

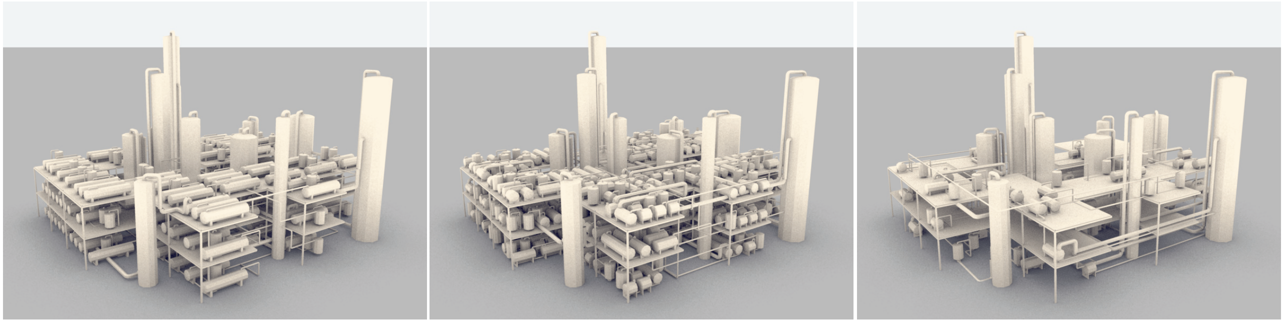
(f) medium cell tessellation
(minimum cell size = 1.4m)



(g) coarse cell tessellation
(minimum cell size = 2m)

(h) without cell tessellation
deviation

(i) with cell tessellation deviation

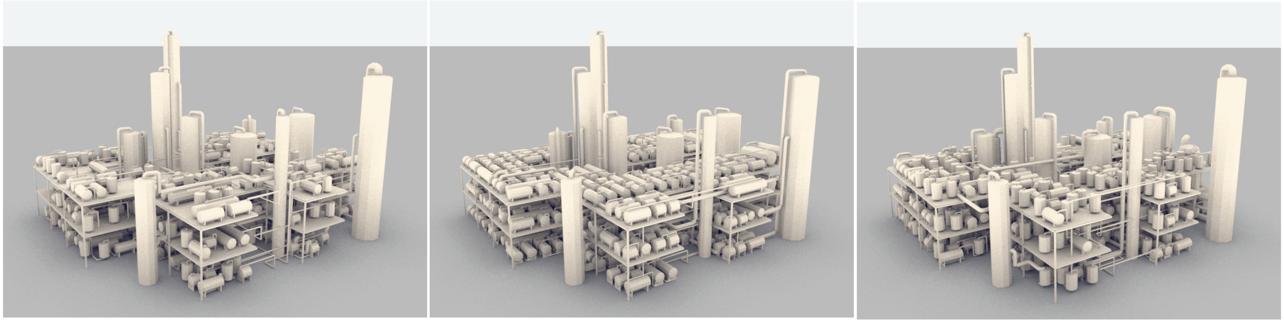


(j) tanks in same direction

(k) tanks in random directions

(l) few tanks

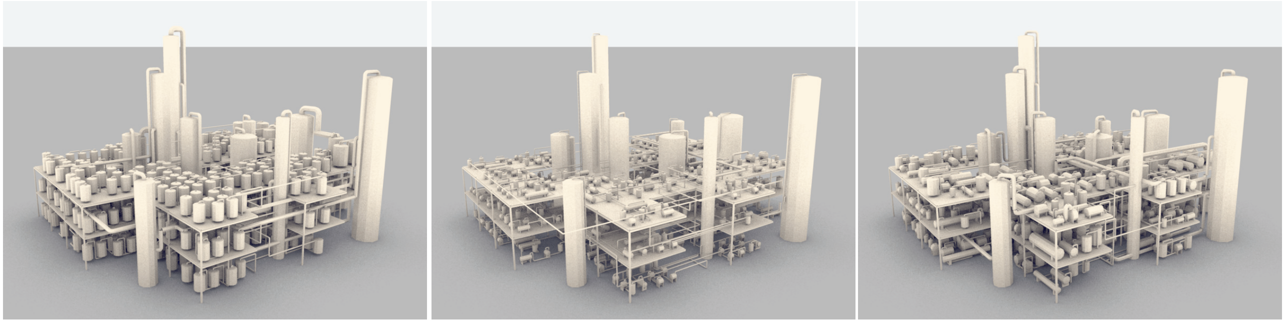
Figure 9: Variations



(m) packed tanks

(n) only horizontal tanks

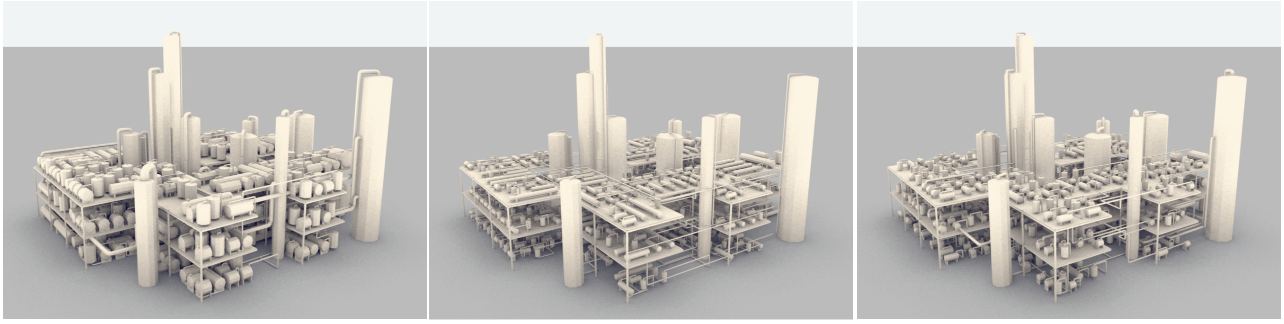
(o) horizontal tanks 50%
vertical tanks 50%



(p) only vertical tanks

(q) only small tanks

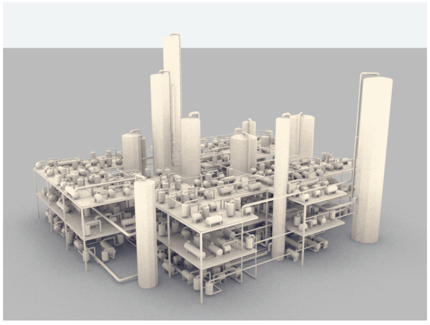
(r) varied tank sizes



(s) only large tanks

(t) only slim tanks

(u) well-aligned tanks



(v) randomly arranged tanks

Figure 9: Variations (continued)