Automated Layout Synthesis and Visualization From Images of Interior or Exterior Spaces

Tomer Weiss*, Masaki Nakada*, Demetri Terzopoulos
University of California, Los Angeles
{tweiss, nakada, dt}@cs.ucla.edu

Abstract

Recent work in computer graphics has explored the synthesis of indoor spaces with furniture, accessories, and other layout items. In this work, we bridge the gap between the physical and virtual worlds: Given an input image of an interior or exterior space, and a general user specification of the desired furnishings and layout constraints, our method automatically furnishes the scene with a realistic arrangement and displays it to the user by augmenting the original image. Our method can deal with varying layouts and target arrangements at interactive rates, which affords the user a sense of collaboration with the design program, enabling the rapid visual assessment of various layout designs, a process which would typically be time consuming if done manually. Our method is suitable for smartphones and other camera-enabled mobile devices.

1. Introduction

The arrangement of objects into a layout is an everyday problem. The problem is involved, because a desirable arrangement may vary greatly according to different use cases, individual styles, and other considerations, while various constraints, such as space bounds, the relationships between different objects, as well as comfort and other functional and aesthetic criteria must be enforced. Layout design is far from trivial for people lacking domain experience, as evidenced by the existence of interior design professionals.

Using broadly available smartphones and other camera-enabled mobile devices, it is easy to share photos of indoor or outdoor spaces and receive suggestions from friends or hired professionals on how to organize and furnish the spaces. Popular consumer mobile applications (e.g., by Amazon) provide limited visualizations of selected furnishings using augmented reality. However, prior work has not addressed important aspects of visualizing spaces of interest that incorporate automatically synthesized suggested layouts. A typical use case would be to provide a computerized alternative to the time-consuming process of manually staging a property for sale or lease through the pleasing layout of furniture and other visible accessory items, which can dramatically influence the perceived property value.

(a) (b)

Figure 1: (a) Image of a vacant living room. (b) Image augmented with a layout synthesized by our method.

Mathematically, layout synthesis yields a challenging non-linear optimization problem. The main goal of our work is to develop a fast, automated system that, given limited user input consisting of a single image of a vacant indoor or outdoor space, visualizes the space furnished with optimal synthesized layouts. Our approach works well with both interior and certain exterior spaces. It is suitable for implementation as a mobile device application constrained by limited computational resources.

The remainder of the paper is organized as follows: Section 2 surveys relevant prior work on layout synthesis as well as on scene understanding from images. Section 3 overviews our algorithmic approach. Section 4 presents our results. There follows in Section 5 a discussion of the limitations of our approach. Section 6 concludes the paper and discusses future work.

2. Related Work

2.1. Layout Synthesis

Layout problems arise in a number of domains. Researchers have applied domain-specific optimization ap-

* T. Weiss and M. Nakada contributed equally to this work.

(a) (b) (c)

Figure 2: Outdoor Yard layout. (a) Random initial state; chairs are colliding unrealistically. (b) Intermediate state of the optimization; not all chairs are facing their respective tables. (c) Final optimized layout; all chairs are in their correct positions and orientations.

proaches to various layouts, from VLSI layouts [30] to architectural floor plans [9, 21]. Layout synthesis also appears in the context of generating virtual worlds for computer games, databases for computer vision algorithm testing, and even virtual reality [33, 13].

Numerous methods have been proposed for synthesizing layouts. Procedural modeling employs grammars [24, 25]. However, these methods require a user to manually encode grammatical rules, a complicated task that is similar to implementing a script for a professional modeling package. Graphical user interfaces help simplify this task [19]. Additionally, modeling grammars may be augmented to interact with external constraints in the form of guidance shapes, user input, or from other models [4, 35, 36].

Optimization-based methods are used to achieve layout goals under a set of predefined constraints. These methods are usually stochastic in nature, sampling layout arrangements from an unknown probability distribution [37, 22, 43, 41]. However, stochastic methods are usually slow when optimizing a layout with dozens of items. Recent work achieves faster running times by combining or using only a continuous, numerical optimization approach [38, 3].

On the consumer side, there exist various software packages and toolkits for designing and visualizing residential layouts [1, 27]. However, these tools require significant manual editing and interior design domain knowledge to achieve satisfactory results.

2.2. Understanding Environments

Before augmenting the environment, ideally one must understand the spatial layout and dimensions [10, 8], the arrangement of objects within the environment [15, 16, 6, 14], the human influence on these arrangements [17], and where the objects should be placed, from small functional objects [16] to evaluating the physical quantities of a layout [45].

To understand a scene directly from an image, Ramalingam et al. [26] proposed a method for deriving the orientation of indoor and outdoor scenes from a single image, combining vanishing points and an optimization procedure that considers all plausible connectivity constraints between lines. In concurrent work, Izadinia et al. [11] propose a system that, given an image, reconstructs an approximate virtual replica of the original scene using a database of CAD models and a deep learning framework. Zhang et al. [44] presented a system for visualizing an augmented indoor scene using a specialized Project Tango tablet, although it does not automatically generate suggestions and requires manual editing.

2.3. Pixel-Wise Semantic Segmentation

Deep learning research has grown dramatically in recent years thanks to algorithmic advances combined with efficient and powerful implementations on GPUs. Most recent results are based on the Visual Geometry Group (VGG) network proposed by Simonyan et al. [32], a very deep network that has produced state of art accuracy in image classification tasks, with various modifications. FCN [20], DeepLab [5], and Dilated Convolutions [42] perform pixel-wise semantic segmentation and have yielded good accuracy for such segmentation problems. The benchmark for semantic segmentation algorithms is the Pascal dataset, which contains images from various domains. Recently, another pixel-wise semantic segmentation algorithm was proposed by Badrinarayanan et al. [2], which also employed a VGG net architecture. In addition to the newly proposed techniques, the network was trained on both a road dataset and an indoor scene dataset provided by SUN-RGBD [39].

Algorithm 1 Automatic Layout Method

```
I ← Get input layout image
O ← Get user layout items and objectives          ▷ 3.3
S ← Segment scene (I)                              ▷ 3.1
S ← Estimate 3D Scene (I)                          ▷ 3.2
S* ← Generate Layout Suggestions (S, O)
I* ← Visualize Augmented Scene (S*, I)             ▷ 3.3
```

3. Algorithm

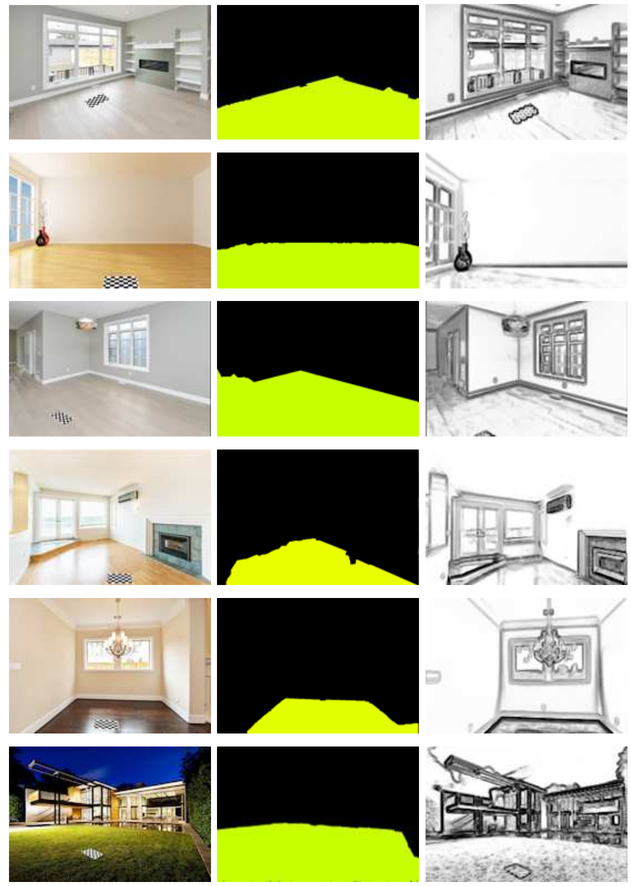
The initial input to our method is an ordinary image of an indoor or outdoor environment. First, we semantically segment the scene depicted in the image, separating the floor/ground from other objects in the scene. Second, we detect the boundary of the floor and other significant edge features. Third, we measure the scale and orientation of the scene by using a checkerboard calibration marker. Finally, we generate an optimal layout with user-selected layout items. Algorithm 1 provides an overview of our approach, indicating the sections in which the details of each step are presented.

3.1. Semantic Segmentation of the Scene

We use a pixel-wise semantic segmentation algorithm to extract the floor/ground. There exist various algorithmic approaches for this pixel-wise segmentation. We chose to use SegNet [2], whose model is trained with a SUN-RGBD dataset [39], mainly because this dataset contains only indoor scene objects while most other datasets have a mix of images from various other categories. Badrinarayanan et al. [2] trained their network with 37 categories, including common layout objects, such as walls, chairs, tables, and the floors of indoor scenes.

For synthesizing layouts, we first must estimate the ground area available. To that end, we need to segment the floor and remove irrelevant objects from the scene. Thus, our task is a binary segmentation problem rather than a multi-label segmentation one. To achieve this segmentation, we add one more layer after the softmax layer of VGG net. This layer takes its output from the softmax layer and produces binary class labels for the floor and other parts of the image. This output is represented by a simple conditional statement where the 'floor' class from the softmax layer produces 'floor', and all of the other labels become 'others'.

Subsequent to the pixel-wise segmentation, we use the approach suggested in GrabCut [29] to retain the floor and remove other components of the scene. Fig. 3 shows the result of this segmentation process. Thus, we detect the location and boundary of the floor or ground, which is necessary in order to establish a layout plane upon which we will synthesize layout items.



(a) (b) (c)

Figure 3: The results of our segmentation and edge detection. (a) Original images. (b) Segmented floors/ground. (c) Edge maps.

3.2. Inferring a 3D Estimate of the Scene

To estimate the size of the room, we ask users to place a checkerboard calibration marker in the scene. The scale of the room is estimated by comparing the known size of the checkerboard to the segmented floor in the scene. The details of this process are as follows:

1. Detect the checkerboard in the scene using Harris corner detection, and compute the room's width, height, and center in pixel coordinates.

2. Define an origin at the center of the checkerboard on the x and y axes of the plane to the horizontal and vertical directions, respectively. The z axis is determined as the cross product of the x and y axes.

3. Employ the checkerboard to calibrate the camera and set up the internal parameters. The camera pose is estimated based on the distance and orientation from the origin. This step is necessary to estimate the position

Figure 4: Synthesized game-room layout. (a) random initial state. (b) Intermediate state of the optimization; layout items are too close for the layout to be deemed comfortable. (c) Final suggested layout with relaxed spacing.

of the virtual camera when rendering the final layout containing the virtual 3D furniture.

4. Traverse the floor geometry from the origin along the x and y axes to compute the distance from the origin to the edge of the floor in pixel coordinates. Since we know the ratio between the size of the checkerboard in pixel coordinates and its real size, we can compute the length of the floor in a scene by scaling its length in pixel coordinates by the ratio obtained from the checkerboard.

5. Apply the Holistically-Nested Edge Detection algorithm [40] to detect the edges of the scene (Fig. 3(c)). This deep-learning-based edge detector results in better edge detection for our indoor scene images than traditional edge detectors, such as the Canny edge detector. We search for edges that are aligned to the z axis, by selecting edge vectors $\mathbf{l}$ whose cosine distance to $\mathbf{z}$,

$$\cos(\mathbf{z}, \mathbf{l}) = \frac{\mathbf{z} \cdot \mathbf{l}}{||\mathbf{z}||\ ||\mathbf{l}||}, \tag{1}$$

is greater than threshold $t = 0.9$. We use the longest edge as the height of the space.

3.3. Layout Synthesis and Visualization

Our layout synthesis scheme is based on a continuous numerical approach to layout synthesis [38], which was inspired by Position-Based Dynamics [23] and by a stochastic McMC scheme for optimizing indoor layouts [43].

Given a layout, the user specifies the furniture items to arrange, and the objectives of that arrangement. The objectives are annotated in terms of geometric constraints, similar to those described in recent work [43, 22]. Among other constraints, our method supports pairwise distance, distance to wall, pairwise and wall rotational constraints, and visual balance, and it can easily be extended with additional constraints. In a typical use case, a user can define the distance between layout items in the same group, together with distance and orientation to the nearest wall. Collisions between competing layout items are automatically resolved. We refer the reader to the above cited papers for the details.

While [38, 43, 22] generate suggestions on a per-object level of detail, we speed up layout generation in some cases by employing a rule-based approach in which each object encompasses a group of layout items. For example, a dining table (Fig. 4) is usually accompanied by a set of matching chairs at prescribed distances and orientations. This is most apparent when furnishing an empty layout. In the common use case, a user who is interested in furnishing an interior layout travels to a nearby retail store and buys furniture items in combinations, as suggested by the retail catalog or displays on site; e.g., a collection that includes sofas, adjoining sofa chairs, and a coffee table.

After obtaining the input layout area, input layout items, and constraints between these items, our method synthesizes the layout, which results in a 3D scene. Given the estimated camera pose, the 3D representation of the layout is rendered into the original image I, to yield the augmented image I^*. For rendering, we use Blender Cycles with the same settings for every scene, except for varying the exposure setting to produce a more realistic result by matching the illumination in I.

4. Results

Our framework is composed of two main components. The semantic segmentation and edge detection component is implemented in Python on an Ubuntu machine with a 3.4 GHz Intel Core i7 and Nvidia Titan X GPU. We use Caffe [12], one of the most common libraries for deep learning in computer vision, for both image processing tasks. Semantic segmentation and edge detection take on average approximately 0.1 seconds and 0.5 seconds, respectively. The layout synthesis component is implemented in Python and Cython. Each layout suggestion is synthesized in no more than 4 seconds. Rendering the final result takes approximately 3 seconds.

We briefly summarize our experiments next:

Hosting Room: (Fig. 1) We used a set of 5 dining tables, a clothes rack, and a floor lamp. We did not impose any strict distance constraints in this setting, except for the floor lamp to be close to the wall.

Outdoor Yard: (Fig. 2) In this scenario, we assigned two patio-style tables with chairs, where the chairs are constrained to be around the tables. We also added a BBQ grill and a trash can. The grill is constrained to be at a greater than minimal distance from the other layout items.

Game Room: (Fig. 4) This scenario includes a sofa, floor lamp, two table tennis tables, and two dining tables for socializing. The sofa is constrained to be near the wall, and the tennis tables close to each other. The sofa is positioned in a slightly rotated position relative to the room's wall for a more comfortable interaction.

Bedroom: (Fig. 5) The bedroom setting included a bed, closet, clothes rack, office table, chair, and a floor lamp. The bed and closet were constrained to be near the wall, and the floor lamp near the table.

Living Room: (Fig. 6) We experimented with a typical living room layout, consisting of a TV, sofa, sofa chairs, and coffee table. We also added an extra table and office chair, and plants. In this setting, we constrained the TV to be the focal point of the furniture groups, consisting of the sofa, sofa chairs, and coffee table. Our system generated 3 different layout suggestions for the same layout items and constraints.

5. Discussion

We have demonstrated the efficacy of our method by augmenting various input scenes, both indoor and outdoor. However, our method has some limitations. An incorrect semantic segmentation of the scene is possible. This typically happens when the input image is not clean; e.g., it contains a large cast shadow, is under-exposed or over-exposed, or the floor has a non-uniform pattern. Fortunately, these are unlikely occurrences for our target use case; i.e., real estate staging, since the sellers of a property tend to capture high-quality images.

In the present study, we did not train our network on outdoor scenes. Therefore, we expect these scenes to be more challenging. Nevertheless, we also tested our method on outdoor scenes (Fig. 2), obtaining some adequate results. We observed that outdoor scenes where similar to indoor scenes with respect to the surface layouts and other segmentation features of the latter. In general, however, our method will not be reliable on outdoor images.

We used a checkerboard to determine the scale of the scene, calibrate the camera, and appropriately set the virtual camera used to render the completed layout. This calibration worked well when the space is approximately rectangular, but it is not accurate in cases where the space has a non-standard shape. Moreover, placing a marker in a scene is not always convenient.

A user of our method must manually assign constraints for the layout synthesis step. Assigning these constraints is straightforward, albeit not automatic. This step can easily be interchanged with a user-friendly set of questions regarding the user's layout preferences.

6. Conclusion and Future Work

To our knowledge, our system is the first complete, interactive system for augmenting images of indoor or outdoor spaces with the highly automated synthesis of furnished layouts. Users of our system can range from ordinary consumers who are looking for a new residence or are interested in remodeling an existing residence, to interior designers and real-estate professionals.

In future work, we plan to improve our system by training it on a more diverse set of images; e.g., by collecting various images, including outdoor scenes and more complex or cluttered interior scenes from the several commonly available datasets [7, 34, 31, 13]. Additionally, to further improve the accuracy of our system, we plan to implement a better scene layout understanding algorithm, such as the one by Ramalingam and Brand [26] that estimates a layout using vanishing lines, or the one by Ren et al. [28] that proposes FCN-based scene layout estimation. We also plan to combine holistically-nested edge detection [40] to improve the accuracy of our layout detection, which should provide us better 3D scene understanding. Better scene understanding would enable an extended version of our system to handle images of spaces containing existing furniture, which would help people who want to add new pieces of furniture or otherwise augment their spaces. Finally, we are interested in improving visual quality by incorporating an effective approach for estimating the lightning conditions in the original image [18] so as to more realistically illuminate the synthesized layouts.

References

[1] Autodesk. Homestyler. http://www.homestyler.com, 2011. 2

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv Preprint arXiv:1511.00561*, 2015. 2, 3

[3] F. Bao, D.-M. Yan, N. J. Mitra, and P. Wonka. Generating and exploring good building layouts. *ACM Trans. Graphics*, 32(4), 2013. 2

[4] B. Beneš, O. Šť'ava, R. Měch, and G. Miller. Guided procedural modeling. *Comp. Graphics Forum*, 30(2):325–334, 2011. 2

(a) (b) (c)

Figure 5: (a) A vacant bedroom. (b) Intermediate layout. (c) Synthesized bedroom.

(a) (b) (c)

Figure 6: Living Room. Our method provides different layout suggestions (a)–(c) for the same collection of furniture items and layout constraints.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *arXiv Preprint arXiv:1606.00915*, 2016. 2

[6] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Indoor scene understanding with geometric and semantic contexts. *Int. J. Comp. Vis. (IJCV)*, pages 204–220, 2015. 2

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes challenge: A retrospective. *Int. J. Comp. Vis. (IJCV)*, 111(1):98–136, 2015. 5

[8] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Adv. Neural Info. Proc. Sys. (NIPS)*, pages 1288–1296, 2010. 2

[9] M. Harada, A. Witkin, and D. Baraff. Interactive physically-based manipulation of discrete/continuous models. In *ACM Trans. Graphics*, 1995. 2

[10] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *Proc. Int. Conf. on Comp. Vis. (ICCV)*, 2009. 2

[11] H. Izadinia, Q. Shan, and S. M. Seitz. IM2CAD. In *IEEE Conf. Comp. Vis. Pat. Rec. (CVPR)*, 2017. 2

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM Int. Conf. Multimedia*, pages 675–678, 2014. 4

[13] C. Jiang, Y. Zhu, S. Qi, S. Huang, J. Lin, X. Guo, L.-F. Yu, D. Terzopoulos, and S.-C. Zhu. Configurable, Photorealistic Image Rendering and Ground Truth Synthesis by Sampling Stochastic Grammars Representing Indoor Scenes. *arXiv Preprint arXiv:1704.00112*, 2017. 2, 5

[14] Y. Jiang, H. S. Koppula, and A. Saxena. Modeling 3D environments through hidden human context. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 2040–2053, 2016. 2

[15] Y. Jiang, M. Lim, and A. Saxena. Learning object arrangements in 3D scenes using human context. In *Proc. Int. Conf. on Machine Learning*, USA, 2012. 2

[16] Y. Jiang, M. Lim, C. Zheng, and A. Saxena. Learning to place new objects in a scene. *Int. J. Robotics Research*, 31(9):1021–1043, 2012. 2

[17] Y. Jiang and A. Saxena. Infinite latent conditional random fields for modeling environments through humans. In *Robotics: Science and Systems*, pages 1–8, 2013. 2

[18] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. *ACM Trans. Graphics*, 30(6):157, 2011. 5

[19] M. Lipp, P. Wonka, and M. Wimmer. Interactive visual editing of grammars for procedural architecture. *ACM Trans. Graphics*, 27(3), 2008. 2

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comp. Vis. Pat. Rec. (CVPR)*, pages 3431–3440, 2015. 2

[21] P. Merrell, E. Schkufza, and V. Koltun. Computer-generated residential building layouts. *ACM Trans. Graphics*, 29(6):181, 2010. 2

[22] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. Interactive furniture layout using interior design guidelines. *ACM Trans. Graphics*, 30(4):87, 2011. 2, 4

[23] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *J. Vis. Communication and Image Representation*, 18(2):109–118, 2007. 4

[24] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. *ACM Trans. Graphics*, 25(3), 2006. 2

[25] Y. I. H. Parish and P. Müller. Procedural modeling of cities. In *Proc. ACM SIGGRAPH'01*, pages 301–308, 2001. 2

[26] S. Ramalingam and M. Brand. Lifting 3D Manhattan lines from a single image. In *Proc. Int. Conf. on Comp. Vis. (ICCV)*, pages 497–504, 2013. 2, 5

[27] D. Reif. *Home Quick Planner: Reusable, Peel & Stick Furniture & Architectural Symbols*. Gardeners' Guide, 1993. 2

[28] Y. Ren, C. Chen, S. Li, and C.-C. J. Kuo. A coarse-to-fine indoor layout estimation (CFILE) method. *arXiv Preprint arXiv:1607.00598*, 2016. 5

[29] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 23(3):309–314, 2004. 3

[30] M. Sarrafzadeh and D.-T. Lee. *Algorithmic Aspects of VLSI Layout*, volume 2. World Scientific, 1993. 2

[31] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. *European Conf. on Comp. Vis. (ECCV)*, pages 746–760, 2012. 5

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv Preprint arXiv:1409.1556*, 2014. 2

[33] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes. A survey on procedural modelling for virtual worlds. *Comp. Graphics Forum*, 33(6):31–50, 2014. 2

[34] S. Song, S. P. Lichtenberg, and J. Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conf. Comp. Vis. Pat. Rec. (CVPR)*, pages 567–576, 2015. 5

[35] O. Šťava, B. Beneš, R. Měch, D. G. Aliaga, and P. Krištof. Inverse procedural modeling by automatic generation of L-systems. *Comp. Graphics Forum*, 29(2), 2010. 2

[36] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. Měch, and V. Koltun. Metropolis procedural modeling. *ACM Trans. Graphics*, 30(2), 2011. 2

[37] C. A. Vanegas, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and P. Waddell. Inverse design of urban procedural models. *ACM Trans. Graphics*, 31(6), 2012. 2

[38] T. Weiss, M. Nakada, C. Jiang, A. Litteneker, N. Duncan, L.-F. Yu, and D. Terzopoulos. Position-based fast layout optimization, 2017. In preparation. 2, 4

[39] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conf. Comp. Vis. Pat. Rec. (CVPR)*, pages 3485–3492, 2010. 2, 3

[40] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proc. Int. Conf. on Comp. Vis. (ICCV)*, pages 1395–1403, 2015. 4, 5

[41] Y.-T. Yeh, L. Yang, M. Watson, N. D. Goodman, and P. Hanrahan. Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Trans. Graphics*, 31(4):56, 2012. 2

[42] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *Int. Conf. on Learning Representations*, 2016. 2

[43] L.-F. Yu, S. K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. Osher. Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graphics*, 30(4):86:1–11, 2011. 2, 4

[44] E. Zhang, M. F. Cohen, and B. Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Trans. Graphics*, 35(6), 2016. 2

[45] Y. Zhu, C. Jiang, Y. Zhao, D. Terzopoulos, and S.-C. Zhu. Inferring forces and learning human utilities from videos. In *IEEE Conf. Comp. Vis. Pat. Rec. (CVPR)*, pages 3823–3833, 2016. 2