SYMPOSIUM ON SIMULATION IN ARCHITECTURE + URBAN DESIGN
SimAUD 2020

A Deep Image of the City: Generative Urban-Design Visualization

Ariel Noyman[1], **Kent Larson**[2]

[1]City Science Group, MIT Media Lab, Cambridge, MA, noyman@mit.edu
[2]City Science Group, MIT Media Lab, Cambridge, MA, kll@mit.edu

ABSTRACT
Streetscape visualizations are necessary for the understanding and evaluation of urban design alternatives. Alongside blueprints and textual descriptions, these design aids can affect city-form, building-codes and regulations for decades to come. Yet despite major advancements in computer graphics, crafting high-quality streetscape visualizations is still a complex, lengthy and costly task, especially for real-time, multiparty design sessions. Here we present DeepScope, a generative, lightweight and real-time platform for urban planning and cityscape visualization. DeepScope is composed of a Generative Neural Network (DCGAN) and a Tangible User Interface (TUI) designed for multi-participants urban design sessions and real-time feedback. In this paper we explore the design, development and deployment of the DeepScope platform, as well as discuss the potential implementation of DeepScope in urban design processes. Demonstration and code are available at: https://www.media.mit.edu/projects/deep-image-of-the-city/

Author Keywords
Urban design; GANs; Generative Design; Visualization; Collaborative Planning

ACM Classification Keywords
I.6.1 SIMULATION AND MODELING (e.g. Model Development). See: `http://www.acm.org/about/class/1998/` for more information and the full list of ACM classifiers and descriptors.

1 INTRODUCTION: THE IMAGEABILITY OF THE CITY

"To understand the role of environmental images in our own urban lives (...) we needed to develop and test the idea of imageability (...) and thus to suggest some principles for urban design." [21]

Urban design renderings and streetscape visualizations are essential for designers, stakeholders and decision-makers during city-design processes. These visual aids can clarify the outcomes of complex design decisions, such as zoning, building codes or land-use allocations, and can affect urban development for decades to come [1, 37]. The importance of understating the impacts of urban design on the street-level was known to architects and planners for centuries, but tools, mediums and techniques to communicate these effects were often limited [5].

Figure 1. DeepScope TUI: Multiple users can simultaneously interact and discuss urban design iterations. The table-top is used as both the design space and a schematic urban top-view. The vertical monitor visualizes the DCGAN street view.

In his seminal 1960 book, Kevin Lynch introduced 'imageability' as a novel approach to visual perception of urban environments [21]. Lynch suggested a toolset for classification of city-form, in which nodes, landmarks, paths, edges and districts reflect the sensation of transitioning through the urban scape. Later, in 'The View From the Road' study [3], Lynch's 'imageability' paradigm was tested using a new medium: Lynch mounted a dashcam to a car, and went on several rides around Boston and other US metros [2]. When later played, these recordings were sped to reflect the overall

SimAUD 2020 May 25-27 Vienna, Austria

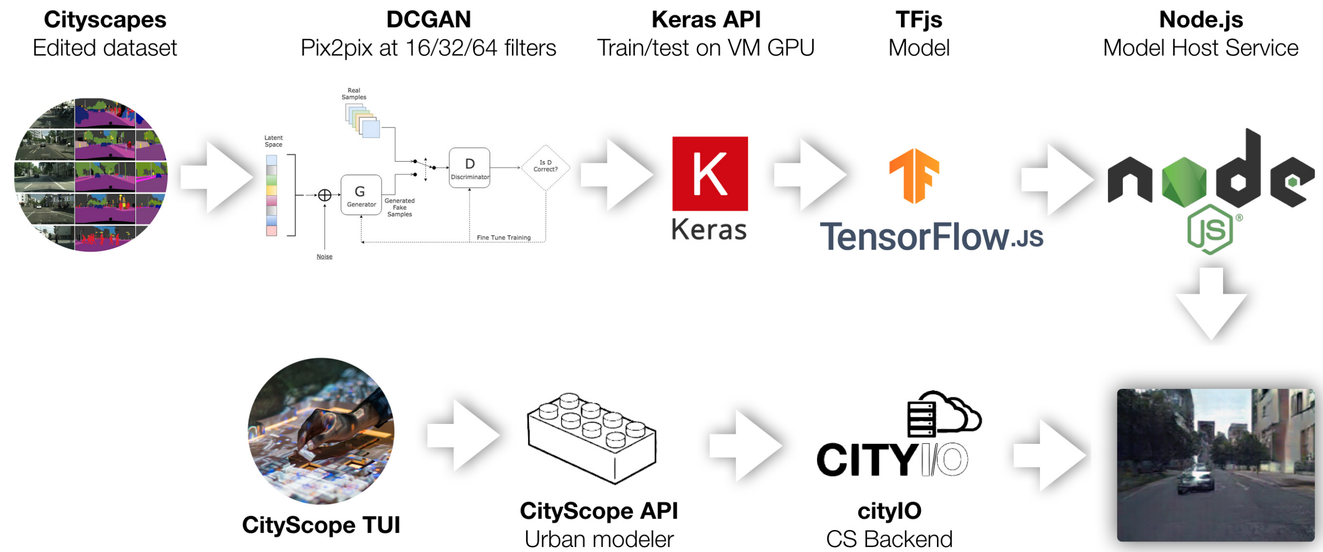

Figure 2. Top row: Model trained on Cityscapes dataset, deployed as node.js app. Bottom row: TUI triggers DCGAN renderings.

'feel' and 'mood' of the road trip; Lynch proposed to overlook fine-grained street elements or the architectural details, and instead focus on the 'imageability' of the urban outline: What is the composition of the built mass? What shapes the street-section? Are there any noticeable landmarks? In the following years, Lynch's innovative documentation techniques became mainstream tools in the field of urban-design [7, 30].

1.1 The Challenge of Visualisation

Despite Lynch's contribution to the perception of cities, documenting existing environments is not sufficient for predicting the impact of future interventions. As Batty concludes, urban visualizations are critical during initial design stages, when the context of the design challenge is only being established, as well as to the generation and evaluation of alternative designs [5]. In the last few decades, advancements in CAD and computer graphics introduced numerous tools to visualize future urban developments [35, 17]. Yet despite their abundance, only few tools offer real-time, realistic urban visualizations during collaborative design processes [26]. Most CAD tools carry complex setups, costly hardware and software, and steep learning curves [39, 22]. These tools often require users to set up many control parameters in virtual environments, such as cameras, lights, materials or shaders. This process might become laborious in complex design scenes, and can gravely affect the outcome, cost and duration of visualization processes [19].

Moreover, common CAD User Interfaces rarely support multi-user collaborative design. This limits decision-makers and stakeholders from taking an active part in iterative design sessions, and forces a synchronous decision making process. Lastly, early urban design stages suffer from lack of design details, which hinders realistic visualizations. These stages commonly involves crude massing exercises, and lack street-level details, so that visualizations are schematic at best [11, 6].

2 DEEPSCOPE: METHODOLOGY AND SYSTEM DESIGN

This paper presents DeepScope, a collaborative, tangible and real-time urban design and visualization platform. DeepScope allows multiple users to collaboratively perform early urban design and land-use allocation sessions, and observe the outcomes as realistic streetscape visuals. Unlike CAD tools, DeepScope offers minimal setup, simple and cheap hardware and software, and requires no expertise to use.

This section details the main parts of DeepScope: (i) a tangible user interface (TUI) for rapid urban prototyping, and (ii) a Deep Convolutional Generative Adversarial Network (DCGAN) for visualizations: As users interact with the TUI, a virtual city model is procedurally updated and fed into the DCGAN model. The model then generates a cityscape visualization based on a user-selected view. The rest of this section explores DeepScope TUI, hardware components, and user interaction.

2.1 HCI Platform for Rapid Urban Prototyping

DeepScope Tangible User Interface (TUI) is built for iterative urban design and land-use allocation. This TUI offers a playful, multi-user tangible environment for design that is augmented by real-time visualization.

Traditional Computer Aided Design (CAD) tools were commonly built around a single user with limited inputs (mouse, keyboard) and outputs (monitor, printer). These interface

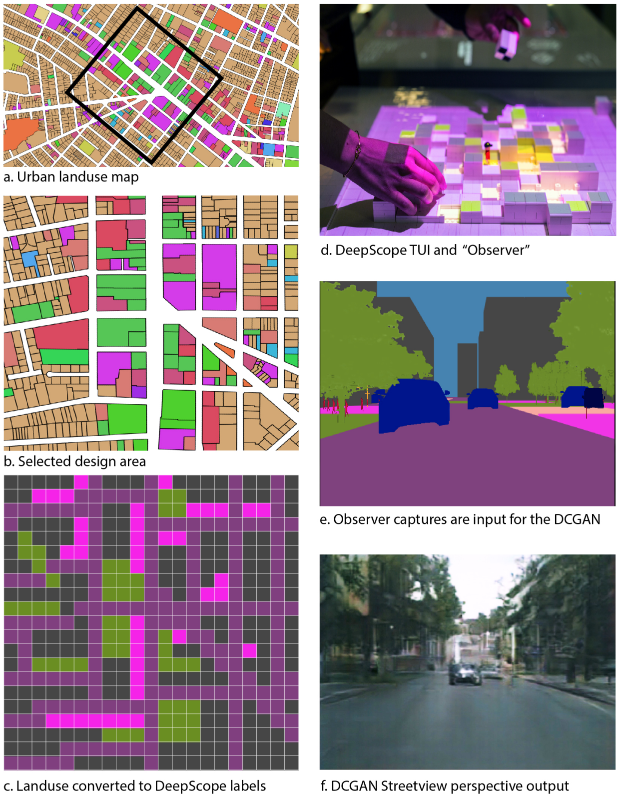

Figure 3. DeepScope process: (a,b) designating an urban intervention site (c) translating the site's land-use/zoning bounds and (d) user-interaction into (e) procedural 3D environment and (e) passing it to DCGAN model for generation of a street-view visualization

Figure 4. Multi-user interaction with DeepScope. Depending on scale and extents of urban context, design sessions can accommodate up to 15 users

Group	**Classes**
flat	road*+; sidewalk*+ parking*+; rail track
human	person*; rider*
vehicle	car*; truck*; bus*; on rails; motorcycle; bicycle*; caravan; trailer
construction	building; wall*; fence; guard rail; bridge
object	pole*; pole group; traffic sign*; traffic light*
nature	vegetation*+; terrain
sky	sky*
void	ground; dynamic; static

Table 1. Cityscapes classes: Marked with 'plus' are labels which can be altered dynamically using CS TUI. Marked with 'star' are labels that are generated dynamically in the 3D model

were not initially conceived as collaborative design tools, even when computer networks became mainstream [38, 5].

In past decades, several TUIs have been developed to facilitate collaborative urban design, augmented by computational analytics. Among these are the Augmented Urban Planning Workbench, the I/O Bulb, The Clay Table and Sensetable [13, 14, 29], all built to allow teamwork and collaboration in urban design processes. In recent years, The MIT City Science group has been developing CityScope (CS): an urban modeling, simulation and collaborative decision-making platform. CS merges TUIs and analytical modules to support a collaborative, evidence-based discourse around the built environment [27]. For the purpose of this research, a CS instance was developed, constructed and tested in an active demonstration space at the MIT Media Lab, Cambridge, MA.

2.2 DeepScope User Interaction

The TUI is composed of three components: (i) a physical urban model, (ii) a scanning module and (iii) a feedback module. The urban model includes an arbitrary grid of tiles, tagged with binary patterns. The tiles are made out of 4x4 LEGO bricks, which were found to induce interaction and creativity during CS design sessions [27]. Each pattern is a 16 bit code of black or white 1x1 LEGO studs, allowing over 65,000 unique pre-defined land uses and attributes. Figure 3.d depicts a user positioning a tagged LEGO brick into the TUI design space.

Each grid-cell pattern represents a different streetscape class: roads, buildings, green-spaces, parking, sidewalks, etc. Each class instance contains additional parameters, such as height, volume, shape, rotation or density. Table 1 specifies the classes and their attributes. When the user shifts a tile, the scanning module detects the interaction through a scanning and networking tool using OpenCV and Node.js. Lastly, a feedback module, containing monitors and projectors, communicates the interaction and analysis outcomes back to the users. This interface has been shown to allow for rapid design iteration, facilitate collaboration and engage users in urban design processes [28].

2.3 Procedural Cityscape Environment

With each interaction, the scanner decodes the new grid-cell patterns and updates the table's data structure. This triggers a regeneration of a virtual 3D environment, in which each grid-cell is represented via its class and additional parameters (see figure 5). As users allocate tiles, the environment is procedurally filled with streetscape elements: a vegetation pattern will create a surface with procedural trees, bushes or live-fences; A sidewalk pattern will produce pedestrians and street-signage, and a parking-lot pattern will be proliferated with parked vehicles. This 3D environment is uniformly hued with RGB values that correspond to input classes expected by the Neural Network model (see section 3). The scanning and 3D scene generation is done on a client-side web-browser using a simple webcam and a WebGL program [25].

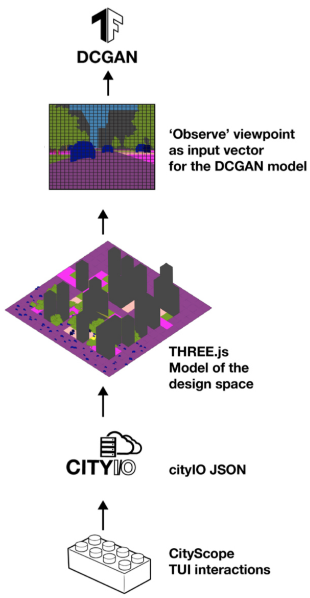

Figure 5. TUI to latent space: TUI interactions are analyzed using OpenCV and streamed as JSON with the webGL app. A 3D model is created based on the JSON array and the Observer viewing angle. Lastly, a snapshot image is fed as an input vector to the DCGAN model.

2.4 Observer

The urban environment designed by the users is constantly 'photographed' by the 'Observer' grid-cell. Similar to Lynch's 'View form the Road' [3], this unique pattern mimics a virtual nomad in the city, and allow users to sets its position, point-of-view and angle. The 'observer' baseline parameters (such as FOV, Frustum and height) were approximated to the camera calibration appendix of the Cityscapes dataset [8]. Additional camera controls were implemented to allow users to move, rotate, pan or zoom the 'observer' by relocating the cell itself and via custom game-pad joystick (see figure 6).

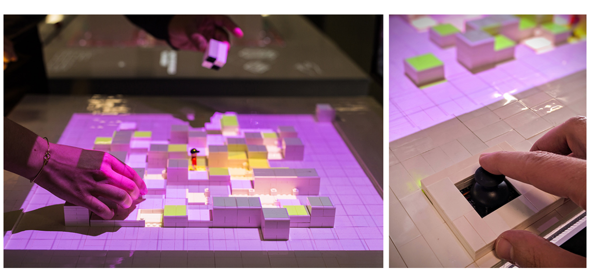

Figure 6. (left) User interaction with grid-cells. (right) 'Observer' viewing angle, depth and position is set via an Arduino Gamepad

2.5 Table-Top Augmentation

The TUI table-top is used as the design space as well as a canvas for visualization. With each design iteration, an illuminated land-use diagram is projected onto the table-top, so that each tile is showing its respective pattern, name or parameters (density, land use, etc.). The Observer position is displayed using perspective cone that indicates its viewing angle and FOV (see figure 7). Together, DeepScope TUI components allow multiple users to design and amend the urban environment and observe the effects of different scenarios on its streetscape.

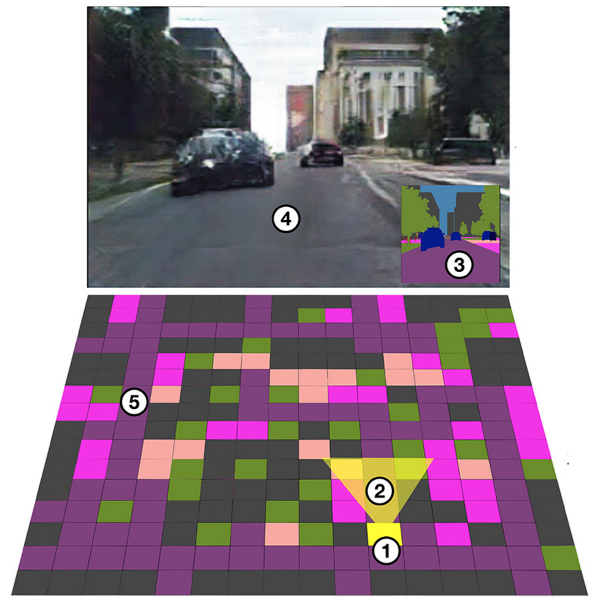

Figure 7. DeepScope TUI: (1) Observer position (2) Observer view angle and FOV cone (3) Observer's 3D street-view as input for DCGAN (4) DCGAN model prediction of street-view (5) TUI interactive grid

3 DEEPSCOPE GENERATIVE NEURAL NETWORK

In order to produce realistic street-view visualization, DeepScope implements a Neural Network (NN) variant called Deep Convolutional Generative Adversarial Network (DCGAN). Following TUI interactions, the Observer's viewpoint is captured and converted into an input vector for the DCGAN (see figure 5). The DCGAN generates an image corresponding to the input vector, where each pixel in the input vector triggers a pixel in the DCGAN output. The resulting image is then drawn onto the DeepScope feedback module. This section explores the dataset, model architecture and NN training.

3.1 Dataset and Model Training

Accurate pattern recognition using NN was already feasible in the late 1980's [18]. However, generating new data that well concatenates a given dataset is still considered a complex problem in Machine Learning [9]. Data generation using NN was greatly advanced with the introduction of Generative Adversarial Networks (GANs)[12]. GANs use two competing NN, Generator and Discriminator, that 'adverse' one another. The Generator attempts to create new data (such as image, sound or text), and the Discriminator aims to nullify these 'fake' creations by comparing them to ground-truth data. The training is completed when the Generator creates indistinguishable samples that constantly fail the Discriminator [15].

3.2 Image-to-Image Translation

A branch of GAN is Conditional GAN (cGAN), in which both NN are given additional data that focuses the generation on specific targets [24, 33]. A notable use-case of cGAN is a pixel-wise conditional generation of images, also known as Image to Image Translation (I2I), or 'pix2pix' [16, 15]. In I2I, pairs of images are used for training, where the pixel values of one image are used as labels (also known as 'classes') of the other. This allows pixel-level prediction using spatial classification of regions in the image [4, 40]. DeepScope implements a lightweight variant of I2I that is fitted for real-time predictions on low-tier devices.

In practice, cGANs extends the classic GAN zero-sum objective function with additional 'class' data: $\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))]$. Here, function V of Generator and Discriminator G, D attempts to minimize a delta between ground-truth data x (in this case, the pixel data) and z, which is the accumulated pixel distribution learnt on each training step (see figure 8). Unlike classic GANs, $\log D(x|y)$ denotes that the additional 'class' data y conditions the learning on both data x as well as on y class. In this respect, distributions created by cGAN generator do not only share resemblance to the learning dataset, but are trained to mimic high-level data structure.

3.3 Cityscapes Dataset

DeepScope's DCGAN model was trained on the Cityscapes dataset [8]. Cityscapes is composed of pairs of street-view images taken using a dashcam around 50 European cities, during different seasons, daytime and weather conditions. Each pair includes a street-view image and a corresponding segmented image with 30 semantic labels. These labels represent different streetscape classes, from buildings and roads to license-plates and road signs. For DeepScope, a pre-processing algorithm was designed to remove motion-blur, increase sharpness, saturation and remove color-casting which were common in a shots taken of a moving vehicle.

3.4 Model Architecture and Performance

DeepScope NN architecture was designed to allow fast predictions, minimal setup and high portability. The Generator has 16 layers with a U-Net [32], encoder-decoder structure. For performance purposes, the Discriminator has 5 layers and is using Leaky ReLU activation that has been shown to improve stability in training [31]. Commonly, DCGAN models benefit from high number of filters set to detect patterns on input data [32]. However, added filters increase the model size, which can gravely impact real-time performance and usability in low-tier devices. In order to still maintain attention to fine details, a shallow NN design with a random up-sampling of 150% was designed [16]. This design allows deployment on most client-side browsers or even on mobile devices, as long as Node.js and TensorFlow.js are supported [36].

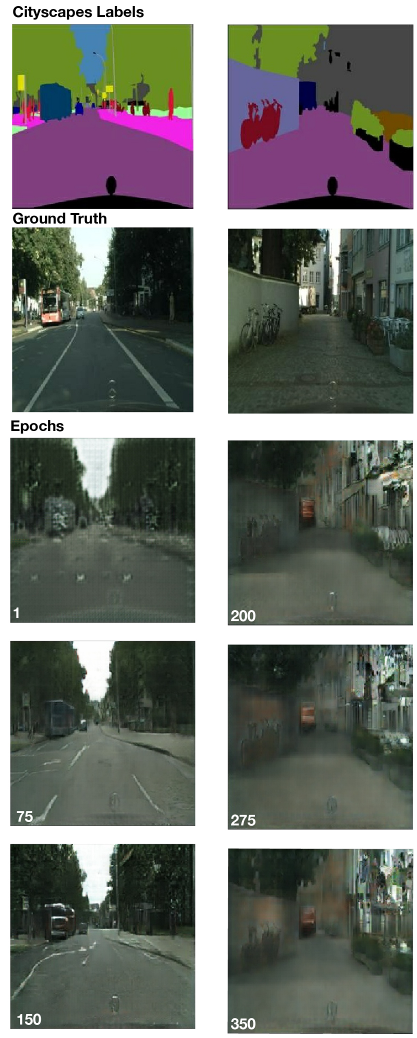

Figure 8. Test samples of different epochs during training: Right column shows quality degradation beyond 200epochs.

3.5 DCGAN Training and Results

As described in Model Architecture and Performance, portability and speed were key factors when balancing between image quality and model size. 20 training sessions were performed with 16,32,64 and 128 filters, with 50 to 2000 epochs. Resulting models were converted to a web format and tested for stability and response time on various client devices. A trained model with 64 filters and 200 epochs showed the best overall results. Models with less filters produced low-quality results; models with 300-2000 epochs demonstrated inconsistencies and 'mode collapse' [4]. Models with more filter were too slow to load and predict in real-time.

3.6 User Interaction Performance Test

In order to avoid interaction latency, two asynchronous processes were used: (i) prediction process and (ii) TUI interaction response. In preliminary user tests, the DCGAN model predicts at ~0.66sec/prediction and the TUI showed a fixed response interval of 50ms. Although the DCGAN slightly trails the TUI, the observation showed that users tend to focus attention to the TUI before expecting the DCGAN output. In that sense, the overall user experience could be considered real-time with continuous design-and-feedback loops [10].

4 DISCUSSION AND CONCLUSION

This paper described DeepScope, a tangible urban design platform for real-time street-view visualization. Visualizations are created using a Deep Convolutional Generative Adversarial Network (DCGAN) trained on the Cityscapes dataset. A tangible user interface for rapid urban prototyping was created for iterations and feedback. The rest of this section will discuss the strengths, weaknesses, threats and opportunities of this work.

4.1 Strengths

DeepScope is designed to allow experts and non-professionals alike to collaboratively experiment with urban design scenarios and real-time feedback. The platform can augment early stages of cityscape design with vivid street-view visuals. These stages have major impacts on urban form and spatial organization of cities, but commonly lack sufficient design representation [5]. Unlike traditional CAD tools, the complexity of creating a 3D urban scene is carried out by DeepScope pre-trained NN. Designed for the web, DeepScope is 'platform-agnostic' and requires minimal computational resources, making it more accessible and portable for public participation. Lastly, the 'unpolished' nature of the GAN outcome allows designers and regulators to focus on the overall 'feel' of the 'Image of the City', instead of highly-specific design details [21].

4.2 Weaknesses

Despite the promise of generative NN, GANs have several drawbacks. First, GANs require large and properly labeled datasets; as such, creating a new Cityscapes dataset for other geographies will involve significant efforts. Several emerging methods suggest decoupled [40] and label-less learning [20], which can simplify the labeling effort. Nevertheless, dataset collection and partial labeling would still be required. Moreover, GANs tend to be inconsistent during learning process, as explored in DCGAN Training and Results [34]. Lastly, the DeepScope GAN would not be able to visualize non-street view angles: Since the Cityscapes dataset was captured using a vehicle dashcam, only matching angles produce reasonable predictions [33]. This issue is common amongst supervised NN, and requires either non-supervised methods or more extensive datasets.

4.3 Threats

The rising popularity of GANs is greatly attributed to their ability to 'create'. Nevertheless, GANs tend to be unpredictable in their results. When it comes to the design practice, certain degree of 'creative freedom' might be desired, yet unpredicted tools might cause resentment or misleading impressions. In the context of DeepScope, the same street-view angle with the same urban-design setup, might produce different visual results if ran twice. While the authors perceive that as a design feature and manifestation of Kevin Lynch's 'Imageability' concept [21], others might observe this as a sign to an untamed technology. Additionally, NN are strictly bounded by their architecture and training data. Tempered NN or datasets can greatly affect the outcomes of the model and inject bias into the results. With machine-learning tools becoming mainstream in the design industry, these concerns should be addressed by testing, validating and open-sourcing design tools, models and data.

4.4 Opportunities

DeepScope can be improved in several aspects: First, emerging NN architectures and training parameters can improve the DCGAN results. Other methods, such as VAE or auto GANs, can produce finer results with greater control [23]. As well, extending the training datasets to different urban environments could yield more versatile representations. Lastly, the TUI can be improved to include multi-scale environments and more finer-grained editing capability.

4.5 Applications and Real-World Implementations

As mentioned in section 2.1, a prototype of DeepScope was constructed and tested in an active demonstration area at the MIT Media Laboratory in Fall '19. During this period, hundreds of design-professionals and random visitors interacted with the tool; their input was incorporated into the UI/UX design and HCI factors of the tool. The most prevailing comments were gathered to appear in this SWOT Analysis.

Currently, DeepScope development is on two major trajectories: *[i] Implementation of DeepScope as a native CityScope module:* The CityScope platform is in the process of adapting a micro-services architecture, in which modules of urban analytics can be 'plugged' into the system when needed. In this context, DeepScope would be used not as a standalone tool, but rather as an additional analysis layer, side by side with other urban matrices (such as noise, ABM, traffic, etc.)

[ii] Real-world testing and deployment: As a standalone tool, DeepScope is now being tested as part of an urban modelling system used in an international urban design competition for a major European city. In the context of this competition, the tool will be used to evaluate street-language and urban form for different design proposals which are at an early schematic design phase. Using DeepScope, both the design teams as well as the jurors would be able to quickly evaluate early-stage urban design decisions, and amend them during sessions. Lastly, the tool might also be incorporated in public participation process in which participants could get better understanding on the implications of different planning alternatives.

More broadly, DeepScope might hint to a future of insightful CAD tools, spanning beyond digital rulers and drafting aids. Such tools would not only expedite tedious tasks, but might be able to leverage the power of advanced computation and become insightful design 'companions'.

REFERENCES

1. Al-Kodmany, K. Using visualization techniques for enhancing public participation in planning and design: process, implementation, and evaluation. *Landscape and urban planning 45*, 1 (1999), 37–45.
2. Andrews, M. The view from the road and the picturesque. In *The aesthetics of human environments*. 2007, 272–289.
3. Appleyard, D., Lynch, K., and Myer, J. R. *The view from the road*, vol. 196. MIT press Cambridge, MA, 1964.
4. Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning* (2017), 214–223.
5. Batty, M., Chapman, D., Evans, S., Haklay, M., Kueppers, S., Shiode, N., Smith, A., and Torrens, P. M. Visualizing the city: communicating urban design to planners and decision-makers.
6. Brusaporci, S. The importance of being honest: Issues of transparency in digital visualization of architectural heritage. In *3D Printing: Breakthroughs in Research and Practice*. IGI Global, 2017, 333–360.
7. Carr, S., and Schissler, D. The city as a trip: perceptual selection and memory in the view from the road. *Environment and behavior 1*, 1 (1969), 7.
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding.
9. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. *IEEE Signal Processing Magazine 35*, 1 (2018), 53–65.
10. Deber, J., Jota, R., Forlines, C., and Wigdor, D. How much faster is fast enough?: User perception of latency & latency improvements in direct and indirect touch. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM (2015), 1827–1836.
11. Drettakis, G., Roussou, M., Reche, A., and Tsingos, N. Design and evaluation of a real-world virtual environment for architecture and urban planning. *Presence: Teleoperators and Virtual Environments 16*, 3 (2007), 318–332.
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Front Matter. *Environmental Fluid Dynamics* (2013), iii.
13. Ishii, H., Ben-Joseph, E., Underkoffler, J., Yeung, L., Chak, D., Kanji, Z., and Piper, B. Augmented urban planning workbench: overlaying drawings, physical models and digital simulation. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, IEEE Computer Society (2002), 203.
14. Ishii, H., Ratti, C., Piper, B., Wang, Y., Biderman, A., and Ben-Joseph, E. Bringing clay and sand into digital design—continuous tangible user interfaces. *BT technology journal 22*, 4 (2004), 287–299.
15. Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. *arxiv* (2016).
16. Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 1125–1134.
17. Kempenaar, A., Westerink, J., van Lierop, M., Brinkhuijsen, M., and van den Brink, A. "design makes you understand"—mapping the contributions of designing to regional planning and development. *Landscape and Urban Planning 149* (2016), 20–30.
18. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation 1*, 4 (1989), 541–551.
19. Lovett, A., Appleton, K., Warren-Kretzschmar, B., and Von Haaren, C. Using 3d visualization methods in landscape planning: An evaluation of options and practical issues. *Landscape and Urban Planning 142* (2015), 85–94.
20. Lucic, M., Tschannen, M., Ritter, M., Zhai, X., Bachem, O., and Gelly, S. High-fidelity image generation with fewer labels. *arXiv preprint arXiv:1903.02271* (2019).
21. Lynch, K. *The image of the city*, vol. 11. MIT press, 1960.
22. Mekni, M., and Lemieux, A. Augmented reality: Applications, challenges and future trends. *Applied Computational Science* (2014), 205–214.
23. Mescheder, L., Nowozin, S., and Geiger, A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org (2017), 2391–2400.
24. Mirza, M., and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
25. Mrdoob. mrdoob/three.js, May 2019.
26. Mueller, J., Lu, H., Chirkin, A., Klein, B., and Schmitt, G. Citizen design science: A strategy for crowd-creative urban design. *Cities 72* (2018), 181–188.
27. Noyman, A. Powerstructures: The urban form of regulation. Master's thesis, 2015.
28. Noyman, A., Holtz, T., Kröger, J., Noennig, J. R., and Larson, K. Finding Places: HCI Platform for Public Participation in Refugees' Accommodation Process. In *Procedia Computer Science*, vol. 112 (2017).

29. Patten, J., Ishii, H., Hines, J., and Pangaro, G. Sensetable: a wireless object tracking platform for tangible user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2001), 253–260.

30. Pearce, P. L., and Fagence, M. The legacy of kevin lynch: research implications. *Annals of Tourism Research 23*, 3 (1996), 576–598.

31. Radford, A., Metz, L., and Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.

32. Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, Springer (2015), 234–241.

33. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in neural information processing systems* (2016), 2234–2242.

34. Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems* (2017), 2990–2999.

35. Shiode, N. 3d urban models: Recent developments in the digital modelling of urban environments in three-dimensions. *GeoJournal 52*, 3 (2000), 263–269.

36. Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., et al. Tensorflow. js: Machine learning for the web and beyond. *arXiv preprint arXiv:1901.05350* (2019).

37. Smith, A., Dodge, M., and Doyle, S. *Visual communication in urban planning and urban design*. University College London, Centre for Advanced Spatial Analysis (CASA), 1998.

38. Sutherland, I. E. Sketchpad a man-machine graphical communication system. *Simulation 2*, 5 (1964), R–3.

39. Yan, J. An evaluation of current applications of 3d visualization software in landscape architecture.

40. Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (2017), 2223–2232.