Sketch Vision: Artificial Intelligence with Sight for Imagination

4.453 Creative Machine Learning
Demircan Tas (tasd@mit.edu)

May 4, 2023

1 Introduction

Figure 1: The Origin of Painting 1786 painting by Jean-Baptiste Regnault (Museum: Museum of the History of France).

Visual design relies on seeing things in different ways, acting on them, and seeing results to act again. Parametric design tools are often not robust to design changes that result from sketching over the visualization of their output. We propose a sketch to 3d workflow as an experiment medium for evaluating neural networks and their latent spaces as a representation that is robust to overlay sketching.

Can state of the art computer vision methods assist designers in seeing? Are they limited to generative uses with predefined surfaces in the design space? Since its emergence, computation in design disciplines created an expectation of alleviating burdens of the design process, either as a smart assistant, creative partner, or a fully automated design tool [1] [3]. The results while convenient in many ways, are short of the full promise [22]. Parametric design results in a paradigm of architects

programming structures via code, yet programming usually follows a process that starts from the completion of traditional sketches and physical models, resulting in these new professionals being less focused on the design, and more on optimization [14]. Grammar based rules are proposed as a method of pruning the myriad of uninteresting options contained in algebras as universes of design [17].

An upcoming paradigm in computation is differentiable programming, exemplified by multi-layer perceptrons, convolutional neural networks, and transformers. Machine learning has been utilized in building technology, and representation of architecture, as well as documentation. Generative AI is addressing the generation of images via prompts, and potential use for early stage design tasks.

Machine learning relies on the definition of domains similar to parametric design, followed by optimizing variables for a desired outcome based on weights acquired via training. Contemporary computer vision overcomes constraints imposed by structures by using pixel representations. Unlike an object oriented representation with defined relationships among constituents, pixels only hold values matched to coordinates. Any relationship among pixels is not defined in advance, but trained using data, in the layers of a neural network. This enables computer vision models to tackle a vast array of tasks from automated medical diagnosis, autonomous vehicles, and even generative systems of art. Computer vision can input an image, and provide a verbal definition, or do the opposite. An image can be generated from another image, and even noise, based on training data [9]. The style of one image can be combined with the content of another [6] [8] [10]. Unsupervised computer vision is capable of detecting concepts that do not readily exist in its own structure, but inferred from data [5].

However, the output distribution in the design space of generative AI is not sufficiently stochastic [9]. Machine learning is inherently limited to the breadth and diversity of examples included in the training set, often leading to biases, which may be perceived as a distinct style in popular examples like Dall-E and Midjourney. Moreover, generative AI tools lack the capacity for generating images with compositional qualities. These models fail when prompted with certain numbers of objects with specific spatial relationships. None of these models can generate "two squares drawn side by side".

1.1 Problem Definition

Parametric design (cite Neil Leach) creates a divide between code, and the form that is generated by it. Code, is an abstraction of the world, blind to its physical manifestation. Sensory apparatus in architecture are developed to feed pre-defined parameters of a system, enabling sensing only as part of an immutable structure.

A major shortcoming of the separation of form and the generative code appear when designers do sketch overlays. It is common in a design context to overlay tracing paper on an existing image, and sketch out design ideas. This approach is in start contrast to the parametric design paradigm, and changes made with sketches commonly lie outside of the design space defined by the parameters of the code. This fundamental difference often necessitates code revisions following design ideation.

We use the form-code duality of parametric design as a model for tackling adaptation in the context of architecture. If a three-dimensional model can be robust against sketches drawn over a rendering of it, we can extend this idea to a system adapting to unforeseen effects of nature.

Differentiable programming is a paradigm that uses neural networks as general function approximators. These functions are not explicitly defined by a programmer, but are derived from data. If the context changes, the system can be fine-tuned by additional data. This approach has been successful for a plethora of tasks in machine learning. Modern computer vision, based on this approach is a prominent domain of study with implications for design and architecture.

We propose a sketch to 3D model based on neural radiance fields (NeRF) as a function that takes two dimensional sketches, and outputs three dimensional models using NeRFs as a representation medium. Neural networks that act as the backbone of this approach are built on parameters as well. However, the parameters of a neural networks are abundant, not pre-defined, and flexible based on training data. A major property of neural networks is differentiability. Differentiable programs can be optimized to approximate any function. Differentiability, paired with a metric for evaluation, affords systems with an ability to *learn* myriads of functions from nature. We aim to evaluate these qualities in an architectural design context to answer the question: Is computer vision capable of seeing like a designer? Moreover can a designer be involved in the process?

2 Related Work

Parametric design is based on programming each step of a design process as distinct objects, and defining chosen variables as parameters that can be manipulated after the design is completed to view alternative results based on unforeseen design constraints. This approach proves useful when the main ideas of the design are already fixed, for adapting the results to small changes in dimensions. Moreover, parameters can be optimized using evaluation metrics. Many alternative designs can be generated and evaluated against a metric, approaching design as an optimization problem in a well defined search space. Methods exist for rapid, and real-time multi-objective design optimization [25].

However, major design changes usually fall outside the scope of changing parameters, and require partial or a complete reconfiguration of the algorithm [23] [4]. Many times, the topology of a three dimensional model dictates its shape, while deformations are possible, topological change requires an application of re-topology [18].

2.1 Cognition, vision, and design

Efforts have been made to explicitly document the activities of making and design [7] [12]. Design activity is carried out through the senses of a designer, creating a relationship with space where the environments acts as an extension of the person, and a memory for design computation [21]. Among the senses of a designer, sight has the role of defining a layer of abstraction, enabling the simplification of complex states presented by the environment. Imaginary lines can use to distinguish common typologies among objects that vary in shape, but share purposes under fixed contexts [16]. Representing the vision of spaces via raster (pixel) representation is a convenient approach for enabling machine learning methods for architecture [19].

2.1.1 Design software disconnect

Despite a body of research in design computation, designers and architects resort to sketches and physical processes when creative solutions are necessary. A disconnect exists between designers and computational tools. Design software depend on designers reasoning like a user, while what designers need is to see like a designer [22] [24]. This results in creative professionals overlaying vellum paper on computer screens or CNC milling scale models to mark desired changes with modeler's tape [26].

3 Methodology

We propose a pipeline for a bidirectional, automated transition between hand drawn sketches, and three-dimensional models (see Figure 2). This transition is divided into two steps, using photographic images as a middle step. We train *Inverse Drawings*, our own model based on [9], [2], to go from line drawings to photographic images. Using [11], we go from photographic images to three-dimensional models. For the reverse direction from three-dimensions to sketches, we use neural rendering to acquire photographic images from NeRFs, and *Informative Drawings*[2] to go from photographic images to pseudo hand-drawn sketches. Out of the four described models for shifting modality, we use three with only minor adjustments.

For sketch to photographic image shift, we train our own model based on the reversal of the architecture proposed for *Informative Drawings*. While there are existing models that go from sketch to photos using Pix2Pix[9], these models rely on image matching and adversarial loss to map sketches to images. Our proposal is based upon the idea that design sketches involve semantic and geometric qualities of three dimensional objects. To reflect this approach, we introduce additional loss metrics to the training process. By implementing *CLIP*[20] into the training process, we add a semantic loss, and by including depth-maps as labels in the training, we use a pre-trained depth inferring model[15] to compute geometry loss.

3.1 Latent Vector Interpolation

Shap-E relies on an encoder-decoder architecture for generating three-dimensional models from images or text prompts. The encoder runs an optimization to locate a million dimensional vector for a matching image. The vector is taken into a forward pass with the decoder to acquire an implicit

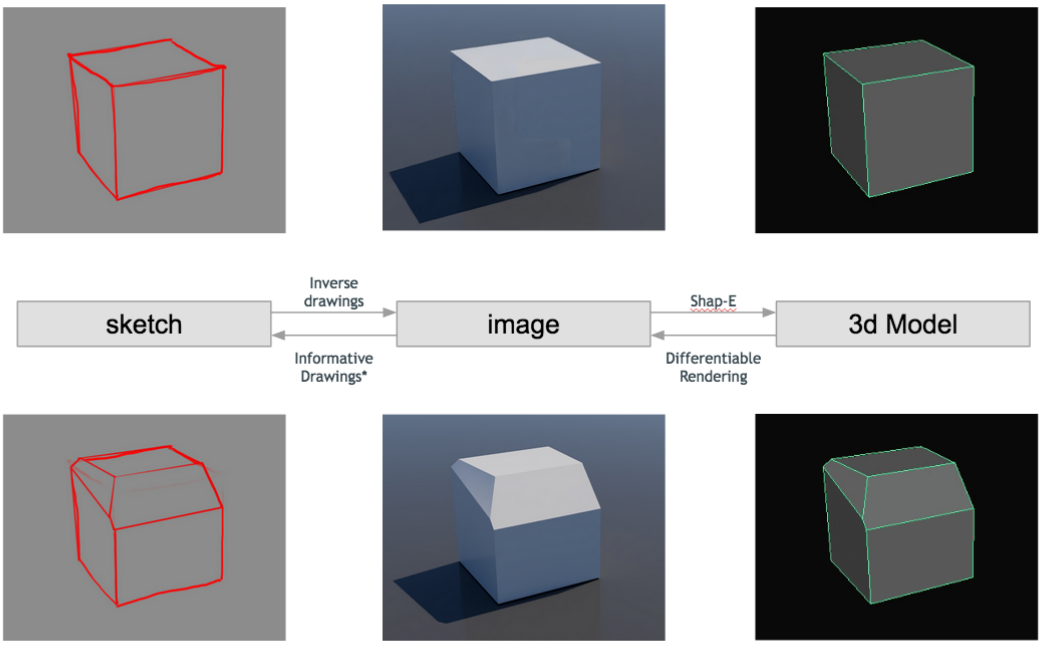

Figure 2: Diagram of Sketch Vision pipeline

Figure 3: Linear interpolation among latent vectors for a chair and a spider

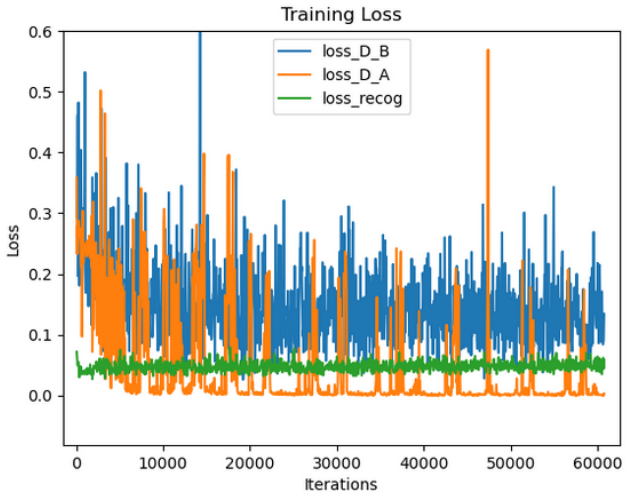

Figure 4: Plot of training losses

Figure 5: Results from Inverse Drawings (sketch created by author)

neural representation. By interpolating among two vectors in n steps, we can pass n acquired vectors into the decoder to generate intermediate objects (see Figure 3).

3.2 Data

We use 8,144 images from the Stanford Cars Dataset[13] for training Inverse Drawings (see Figure 4 for loss plot). We acquire sketches for training input, and depth maps as labels by using Informative Drawings. Moreover, we shuffle photographs from the data-set to use for computing style loss. For the second phase of our experiments, we trained the model with the same scheme, but using 8,144 images from ShapeNet Render[27] data-set. When running Informative Drawings, we used the provided pre-trained models from[2] and [15].

4 Results

We present the isolated results from *Inverse Drawings*, as well as the holistic results of the whole Sketch-Vision pipeline. Due to the ambiguous nature of the sketch to three-dimensions process, we provide qualitative evaluations. When trained on Stanford Cars Dataset, *Inverse Drawings* exhibited strong generalization. Using design sketches of fictional wheeled vehicles as input, the output quality was similar to in domain, production cars. We observed a decrease in quality with sketches of robots and flying ships, and a greater decrease with sketches by the author. Moreover, sketches with greater accuracy tend to yield better results.

Combining the output of *Inverse Drawings* with Shap-E, we observed that the output style of Inverse Drawings was incompatible with the expected input of Shap-E. Since Shap-E was trained on synthetic data with white backgrounds, it does not generalize to input images with non-white backgrounds. By training *Inverse Drawings* on ShapeNet Renders, a data-set of synthetic models rendered in Blender, we achieved a better match (Figure 7). The latter data-set lacks the diversity

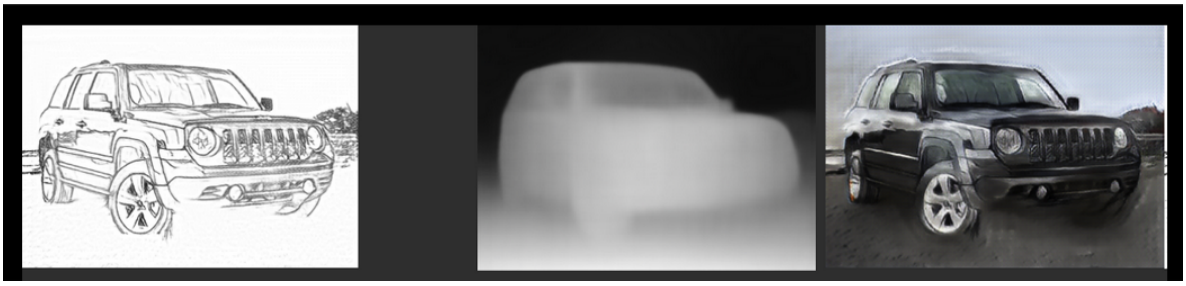

Figure 6: In-domain result from Inverse Drawings

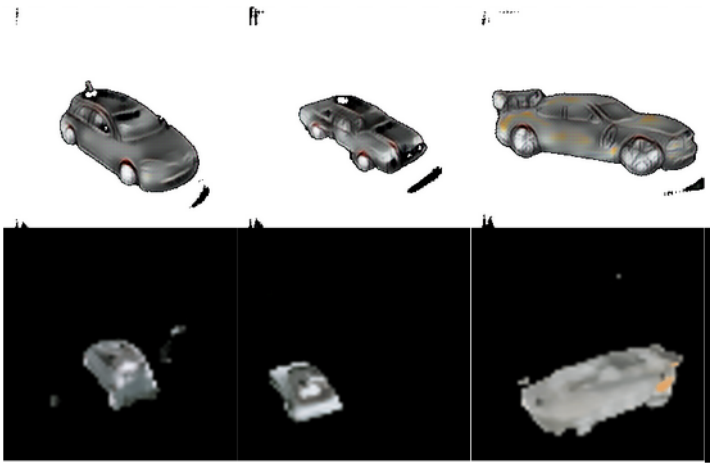

Figure 7: Shap-E with photographic input from Inverse Drawings

of the former, resulting in a lowered capacity to generalize. These constraints were not only in the domain of objects, but also in lighting conditions, camera angles, and material qualities.

5 Conclusion

We present *Sketch-Vision*, a pipeline for automated modality shifts among hand drawn sketches, photographic images, and Neural Radiance Fields. By implementing digital images of sketches as an input, we provide a robust interaction scheme that generalizes to a breadth of use cases, from napkin sketches to digital sketch-overs. Moreover, renders or photographs from existing three-dimensional objects can be *sketchified* to achieve a malleable medium for design modification.

Inverse Drawings generalizes to a limited capacity when trained on rich data-sets with diverse images from the physical world, and works with other models of our pipeline to output three-dimensional representations. By combining a large model trained on a massive, yet domain-specific data-set, with our own model, we achieve an ability to steer said model with our own. As long as our training data is balanced between diversity, and the visual style of *Shap-E*, we achieve representative results in three-dimensions (see Figure 8).

By running the pipeline with multiple sketches, or iterations of a single one, we can acquire latent vectors for each, which we can interpolate in a manner similar to parametric design (Figure 9).

Figure 8: End-to-end example of Sketch Vision (sketch created by author)

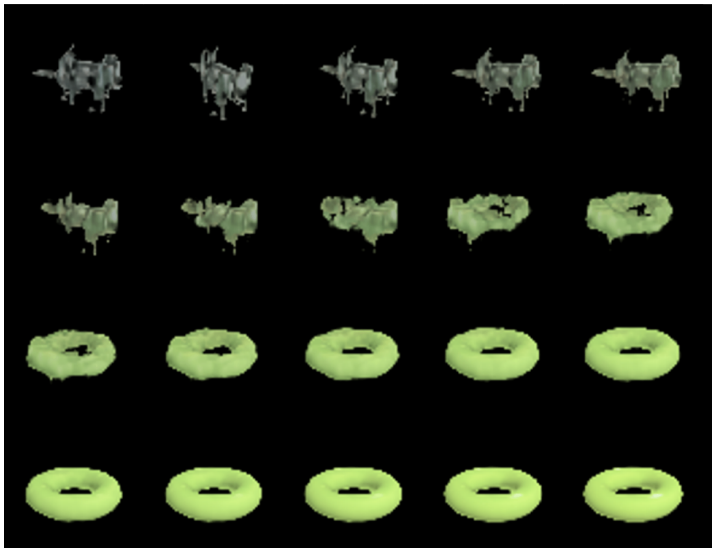

Figure 9: Linear interpolation among author's design and the image of a torus

References

[1] Aish, R. First build your tools. *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design* (2013), 36–49.

[2] Chan, C., Durand, F., and Isola, P. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 7915–7925.

[3] Coons, S. A. An outline of the requirements for a computer-aided design system. In *Proceedings of the May 21-23, 1963, spring joint computer conference* (1963), pp. 299–304.

[4] Dossick, C. S., and Neff, G. Messy talk and clean technology: communication, problem-solving and collaboration using building information modelling. *The Engineering Project Organization Journal 1*, 2 (2011), 83–93.

[5] Forsyth, D. A., and Ponce, J. *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

[6] Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2414–2423.

[7] Gürsoy, B., and Özkar, M. Visualizing making: Shapes, materials, and actions. *Design Studies 41* (2015), 29–50.

[8] Huang, X., and Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 1501–1510.

[9] 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), pp. 1125–1134.

[10] Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision* (2016), Springer, pp. 694–711.

[11] Jun, H., and Nichol, A. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463* (2023).

[12] Knight, T., and Stiny, G. Making grammars: from computing with shapes to computing with things. *Design Studies 41* (2015), 8–28.

[13] Krause, J., Deng, J., Stark, M., and Fei-Fei, L. Collecting a large-scale dataset of fine-grained cars.

[14] Llach, D. C. *Builders of the Vision: Software and the Imagination of Design*. Routledge, 2015.

[15] Miangoleh, S. M. H., Dille, S., Mai, L., Paris, S., and Aksoy, Y. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 9685–9694.

[16] Minsky, M. *Society of mind*. Simon and Schuster, 1988.

[17] Mitchell, W. J. *The logic of architecture: Design, computation, and cognition*. JSTOR, 1990.

[18] Pan, Z., Di, M., Zhang, J., and Ravi, S. Automatic re-topology and uv remapping for 3d scanned objects based on neural network. In *Proceedings of the 31st International Conference on Computer Animation and Social Agents* (2018), pp. 48–52.

[19] Peng, W., Zhang, F., and Nagakura, T. Machines' perception of space: employing 3d isovist methods and a convolutional neural network in architectural space classification.

[20] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning* (2021), PMLR, pp. 8748–8763.

[21] Sass, L., Chen, L., and Sung, W. K. Embodied prototyping: exploration of a design-fabrication framework for large-scale model manufacturing. *Computer-Aided Design and Applications 13*, 1 (2016), 124–137.

[22] Steinfeld, K. Dreams may come.

[23] Stiny, G. *Shape: talking about seeing and doing*. MIt Press, 2006.

[24] Sutherland, I. Structure in drawings and the hidden-surface problem. *Reflections on computer aids to design and architecture* (1975), 73–85.

[25] Terzidis, K. *Algorithmic architecture*. Routledge, 2006.

[26] Verlinden, J., Kooijman, A., Edelenbos, E., and Go, C. Investigation on the use of illuminated clay in automotive styling. In *6th International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID&CD), Delft, NETHERLANDS* (2005), Citeseer, pp. 514–519.

[27] Xu, Q., Wang, W., Ceylan, D., Mech, R., and Neumann, U. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS* (2019).