Exploring Cartoon Priors for Neural Radiance Fields

Joonho Kim

Abstract—Neural Radiance Fields (NeRFs) have gained much traction and success reconstructing novel views of real-world scenes given sparse imagesets, yet the reconstruction of creative input domains such as cartoon images have been less explored. Reconstruction of hand-created cartoon inputs can facilitate design exploration and animation by reducing repetitive drawing tasks for artists. In this project, we explore 3D reconstruction of cartoon image inputs using NeRFs by applying cartoon domain-specific regularization.

Index Terms—Animation, Neural Implicit Representations, Cartoons, 3D Reconstruction

1 Introduction

When creating hand-drawn animation, an animation team will often create a character reference sheet, or a blueprint documenting standard views of important visual features of a character. This blueprint will often display full body views of a character from different camera directions (front, side, back...) and sometimes detailed views of the face, equipment, and body parts. The character reference sheet serves as a baseline guide for drawing a character with consistent style and emotional expression across an animation team.

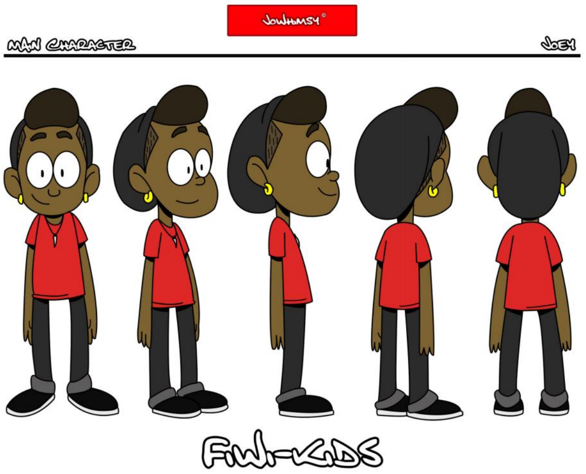

Fig. 1. Cartoon Turnaround of Joey by JoWayne McFarlane: https://www.artstation.com/jowhimsy

Though a character reference sheet provides basic visual guidance, it cannot be used directly to create new character views; the artist cannot use the character reference sheet to directly synthesize a new view. Instead, an artist must imagine the new view mentally and create that mental image onto paper through drawing while using the character reference sheet as stylistic guidelines. This process of transferring imagination to paper through drawing is one of the most time consuming processes of animation.

Animation studios have used computer graphics technology to alleviate this problem by representing characters as 3D models. By creating digital character reference sheets, artists can easily copy these assets into their workstation and render views by toying with digital parameters (curve control points, rigs, and cameras), and as a result most of the time consuming work changes from hand drawings to controlling digital dials. Though widely prevalent in TV and film, the genre does not capture the stylistic qualities of hand-drawn animation: cartoon deformations, emphasized expressions, and cartoon-like motions. Artists prefer carefully hand-crafting characters in 2D because it provides the most amount of view-dependent control 3D models can't provide, but, again, this is a time consuming process.

Recently, generative learning models have produced successful results in novel view synthesis. Neural Radiance Fields have become a new way to take limited views and learn an implicit representation that can generate novel views. However, little work has been done to explore the cartoon domain. The cartoon space has unique rules in design and physics that do not comply in our real-world space.

We explore the NeRF architecture with a restricted cartoon color palette and discuss issues with NeRF using images in the cartoon domain. This paper is meant to spark future work in applying more deep learning methods in the current artist workflow and by no means presents state of the art results.

2 Related Work

Novel view synthesis tackles the problem of rendering any desired view of a subject given only a subset of views and respective renders. Recently the volumetric-based representation Neural Radiance Fields [1] has gained much popularity by representing a subject as a density field and view-dependent radiance. Much follow-up work has driven NeRF's to handle deformations [2], faster representations [3], and fewer input [4].

- *Joonho Kim is with the Department of Computer Science, University of Toronto, Toronto, CA.*
E-mail: joonho@dgp.toronto.edu

Novel view generation of cartoon images within the 2D graphics community have taken more geometric approaches such as with rotating planar representations [5] or image feature matching/interpolation [6]. The most widely sought after application of novel view generation is inbetweening (drawing frames inbetween two key frames) because inbetweening takes up so much time. However, again, representing cartoons with 3D geometry restricts the expressiveness of cartoon-style drawings. Rather than having an explicit 3D geometry, what if we use Neural representations to express our character?

2.1 Cartoon Domain Restrictions

Altering the NeRF architecture to cater cartoon inputs is still very challenging and unsolved. Some notable characteristics of the cartoon domain are the following:

2.1.1 Few Input Views

Character reference sheets often contain less than 8 views of a character which diminishes any hope for modern NeRF's to learn meaningful representations for novel view synthesis. Methods to pretrain NeRFs for feature learning [7] or smoothness regularization [4] attempt to perform few-shot synthesis, but these methods rely on geometric priors which cartoons somewhat inconsistently follow.

2.1.2 View-Dependent "Geometry"

Cartoon character drawings should not only look recognizable but exhibit cartoon-like features. When looking at Mickey Mouse from the front and side view, Mickey Mouse's ears will pivot from the top of his head to the side because the artist could portray his mouse-like features. Or when looking at a side profile of a character, the mouth will often shift closer to the cheek rather than realistically to the side 1. Deformable NeRF's [2] [8] map view dependent inputs to a canonical space (T-Pose for character animation) and learn deformation functions, but these methods assume geometric consistency. Cartoon animators exaggerate characters for cartoon effects (squash and stretch) which don't maintain volume.

2.1.3 Undetermined Camera Matrices

Characters aren't rendered given a camera view matrix but by the artists intuition and pose description. Therefore assigning a camera view matrix to views on a character reference sheets is non-trivial. We can somewhat tackle this issue by asking artists to draw on-top of 3d mannequins where the camera view is known and train the NeRF with these view/matrix pairs, but artistic drawing errors (or exaggerations) render simple camera systems naught.

2.1.4 Flat Shading

Many NeRFs (or any 3D reconstruction tool such as COLMAP) rely on the granularity of an the image signal to make accurate feature locations or loss update (photometric loss). However, flat shading weakens these priors for vision tasks. We could possibly enforce color smoothness using a dirichlet energy.

2.1.5 Black Countour Lines

Unlike real-world scenarios, cartoons often add black contours to outline drawn segments. These contours complicate 3D reconstruction because these contours appear only at locations where the surface normal is perpendicular to the camera. There has been work to generate these suggestive contours [9] to stylize 3D geometry, but performing the reverse task requires consistent stroke geometry and priors on line drawings [10] [11].

3 Proposed Method

3.1 NeRF Overview

The neural radiance field (NeRF) is continuous, volumetric scene represented by the function $F : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ where a 3d position $\mathbf{x} = (x, y, z)$, 2D viewing direction $\mathbf{d} = (\theta, \phi)$, color value $\mathbf{c} = (r, g, b)$, and volume density σ. Essentially, given a 3D point and a viewing direction, NeRF learns a volume density and view-depented color value for each position in the scene.

To render a scene from a radiance field, NeRF uses differentiable volumetric rendering techniques. Given a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, we can render a color $C(\mathbf{r})$

$$C(\mathbf{r}) = \int_{t_c}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}, \mathbf{d})dt$$

$$\text{where } T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)$$

Volume density $\sigma(x)$ represents the differential probability of a ray $\mathbf{r}$ terminating at location $\mathbf{x}$, $C(\mathbf{r})$ is the expected color of ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, t_n and t_f are near and far bounds, and $T(t)$ represents the probability that ray r travels to t from t_n.

In the vanilla NeRF representation, F is a large MLP with multiple layers. To train the vanilla NeRF, we shoot rays into our volumetric scene and sample points $\{x_1, ...x_n\}$ along each ray. Each sampled point x_i is then passed through the MLP $F(x_i)$ to return $(\mathbf{c}, \sigma)_i$ where sample point colors and densities along each ray are combined to produce a color using the volumetric rendering technique. These generated colors are then compared against ground truth colors.

3.2 Color Palette Labels

Most cartoon characters are created with a limited color palette, however NeRFs predict pixels from a continuous colorspace. We can enforce a limited color palette by extracting n major colors in an image and adding a decoder which maps a rendered color to a color label.

Specifically, given an image I, we extract n color labels $L = \{l_1, l_2, ..., l_n\}$ where l_i constitutes at least $\rho = 5\%$ of pixel values in I. We label $C(\mathbf{r})$ to the closest l_i using a small color palette decoder F_c where our distance metric is the L2-norm of RGB vectors and our loss is cross-entropy. Our rendered pixel value for a ray r is $F_c(C(r))$.

The loss function can be altered to become the sum of photometric loss on the original rendered image and cross-entropy loss on the palette-labeled image.

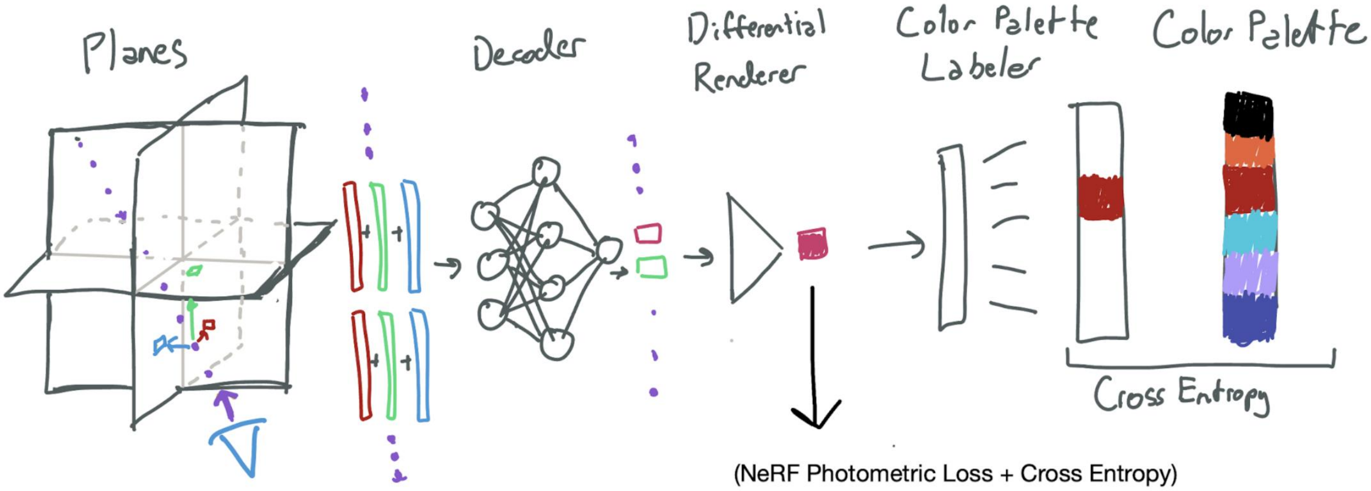

Fig. 2. Architecture: We use a triplanar feature grid inspired by EG3D [12] as our neural representation. Sampled points are projected onto each feature plane to extract a bilinearly interpolated feature vector per plane. These feature vectors are aggregated and fed into a small decoder model to extract RGB+density vectors. We then differentiably render to extract a color value for each ray. Our contribution is appending a color classifier (small MLP + softmax) to map a range of colors to a color palette color label and trained using cross entropy. This color palette can be generated (extcolors) or given. The rendered image pixels use these color palette labels rather than the rendered colors.

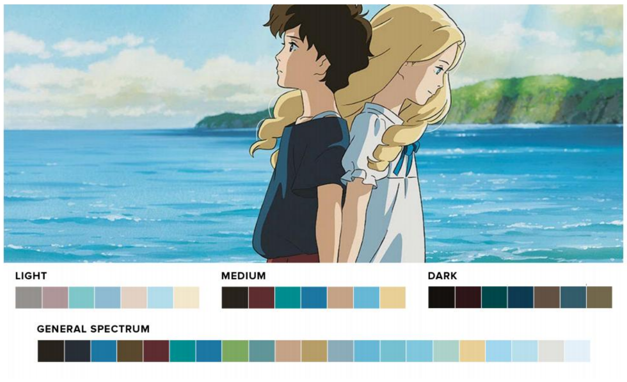

Fig. 3. *When Marnie was There* color palette by Roxy Radulescu of Movies In Color: www.moviesincolor.com

3.3 Implementation

Our input is pairs of character images and camera matrices. Camera matrices were estimated by overlaying images on top a 3D model in bender and tuning the blender camera settings to fit the image. Our color palette was extracted automatically using ExtColors or manually (mimicking an artist providing the color palette). Pairs of image/camera-matrix are fed into a normal NeRF implemented using NVidia's Kaolin Wisp [13] with a custom implementation of a Tri-planar NeRF. Rendered pixels are fed into a small MLP classifier (1 layer, 128 width) and trained using a sum of NeRF's photometric pixel loss + Color Palette Cross Entropy loss as if the color palette is a regularizer with the photometric loss. We use an NVidia Titan RTX with 24GB for our experiment.

4 Experimental Results

We ran our model on 150x150 pixel characters. Our results 4 show that learning from a set color palette improves image quality over using a regular NeRF. With cartoon input and the characteristics in the cartoon-domain mentioned before, the setups do pretty well reconstructing the training images; however our technique works better reconstructing the inbetween than the regular NeRF. We can see the palette classification fills in the face much better than the no palette technique which confuses the face color with the hair color. In our results, we also display renderings with the hand-picked palette technique but before color classification to see if our color-palette classifier MLP was contributing to learned geometry, but we can see that renderings before the color palette classification show the filled in face just as well. The color-palette classifier seems to contribute to the NeRF portion learning geometry.

TABLE 1
Image Evaluations Fig 4

Setups	PSNR	SSIM	LPIPS
No Palette	22.51	0.778	0.303
No Palette (200 iters)	19.878	0.762	0.35
Palette (automatic)	23.18	0.762	0.246
Palette (hand selected, 200 iters)	17.9	0.767	0.271
Palette (hand selected)	**24.91**	**0.802**	**0.21**

5 Conclusion

In conclusion, we present a project on the exploration of Cartoon Domain using a simple color palette classification. We also explain different issues with implementing a NeRF with cartoon inputs and present some different ideas to keep in mind. We hope that the domain of artistic applications will be kept in mind while researching different neural representations such as recent stable diffusion or deformable NeRFs.

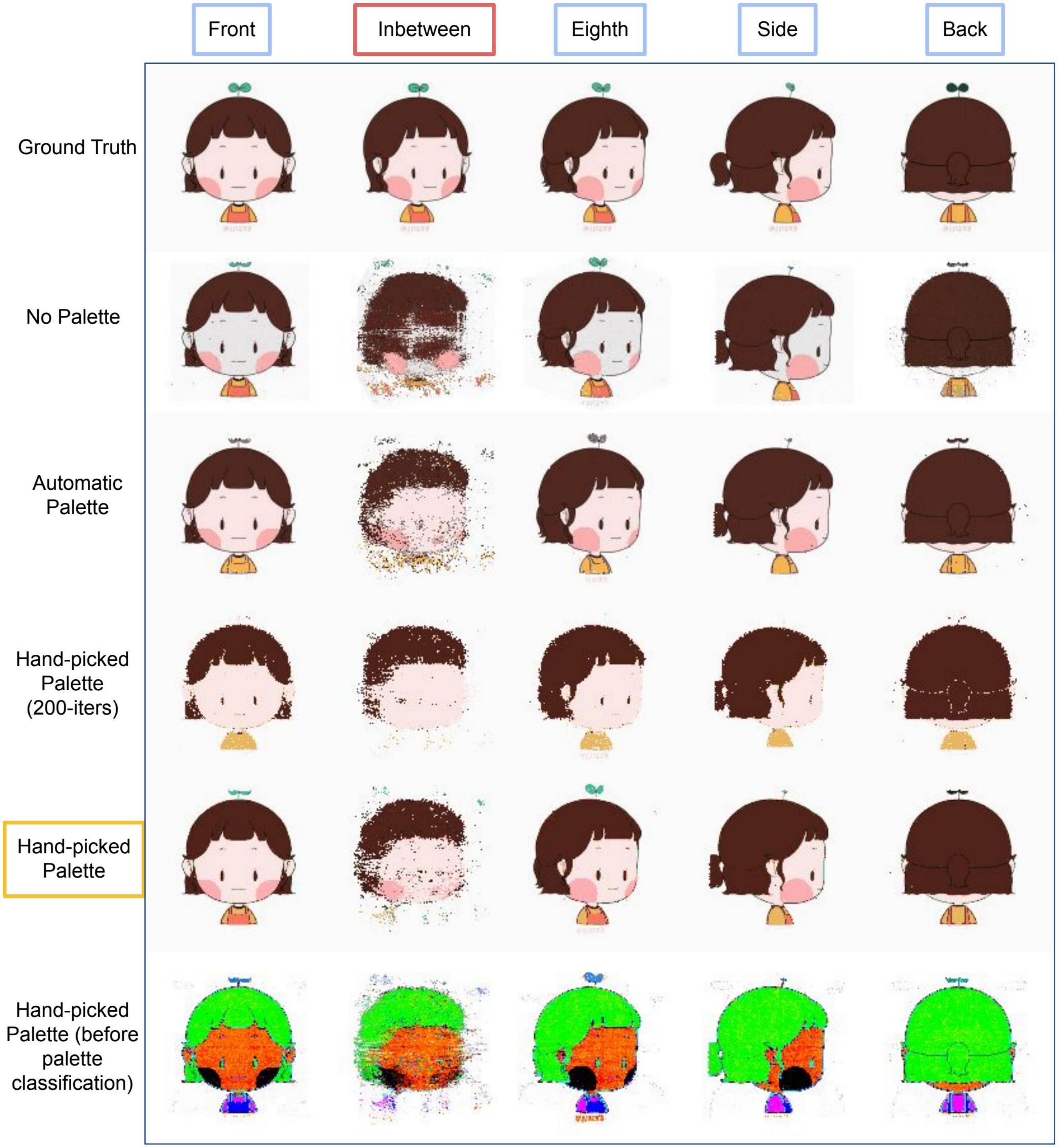

Fig. 4. Rows indicate different setup parameters with our method as the yellow row. Blue columns display training view renderings and red columns display novel view renderings. These experiments were done with 8 training images at 150x150 pixel resolution with 2000 training iterations (6 min). Character Illustration is from Claire J (@jji2yo).

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[2] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874.

[3] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," *arXiv preprint arXiv:2203.09517*, 2022.

[4] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5480–5490.

[5] A. Rivers, T. Igarashi, and F. Durand, "2.5 d cartoon models," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–7, 2010.

[6] H. Choi and S. Lee, "Novel view synthesis from two cartoon face drawings," in *ACM SIGGRAPH 2017 Posters*, 2017, pp. 1–2.

[7] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural radiance fields from one or few images," in *CVPR*, 2021.

[8] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural Radiance Fields for Dynamic Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[9] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, "Suggestive contours for conveying shape," in *ACM SIGGRAPH 2003 Papers*, 2003, pp. 848–855.

[10] E. Yu, R. Arora, J. A. Bærentzen, K. Singh, and A. Bousseau, "Piecewise-smooth surface fitting onto unstructured 3d sketches," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530100

[11] M. Bessmeltsev, W. Chang, N. Vining, A. Sheffer, and K. Singh, "Modeling character canvases from cartoon drawings," *Transactions on Graphics (2015)*, vol. 34, no. 5, 2015.

[12] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. D. Mello, O. Gallo, L. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein, "Efficient geometry-aware 3D generative adversarial networks," in *arXiv*, 2021.

[13] T. Takikawa, O. Perel, C. F. Tsang, C. Loop, J. Litalien, J. Tremblay, S. Fidler, and M. Shugrina, "Kaolin wisp: A pytorch library and engine for neural fields research," https://github.com/NVIDIAGameWorks/kaolin-wisp, 2022.