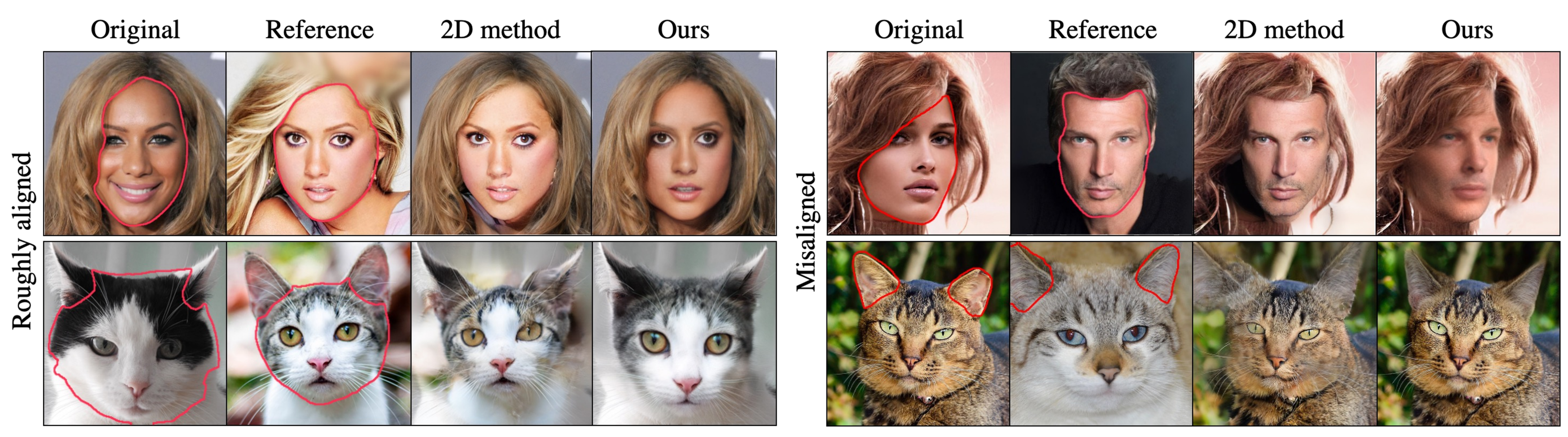
BlendNeRF: 3D-aware Blending with Generative NeRFs

Hyunsu Kim[1] Gayoung Lee[1] Yunjey Choi[1] Jin-Hwa Kim[1,2] Jun-Yan Zhu[3]
[1]NAVER AI Lab [2]SNU AIIS [3]CMU

Project: https://blandocs.github.io/blendnerf
Code: https://github.com/naver-ai/blendnerf

Abstract

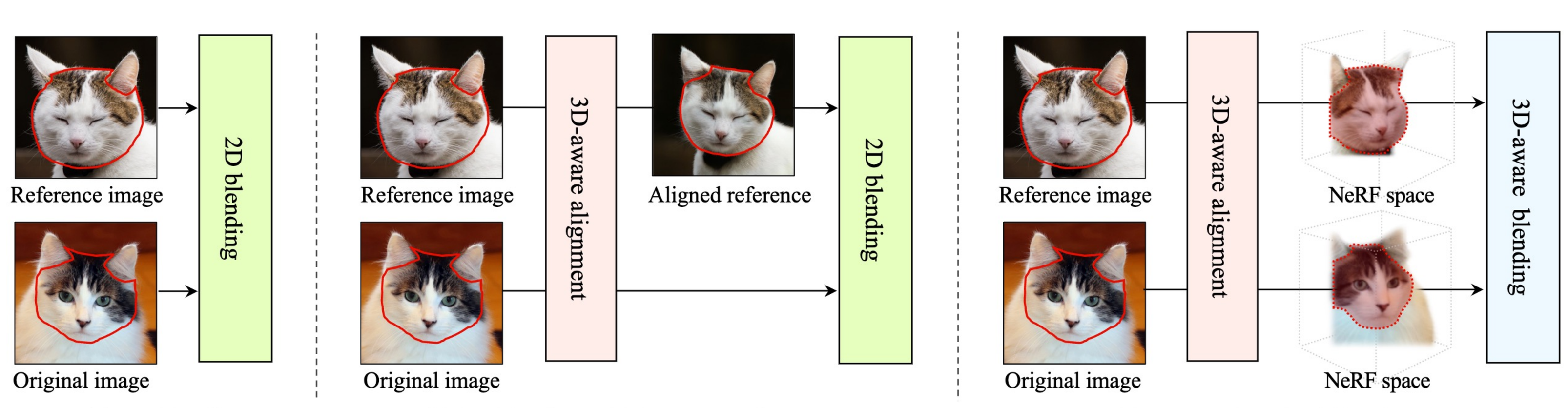


Challenge: Seamless blending of images, especially with misalignments from camera poses and object shapes

Solution: 3D-aware blending using generative Neural Radiance Fields (NeRF)

- **3D-aware Alignment:**
  - Estimate camera poses of the input images
  - Perform pose alignment for objects
- **3D-aware Blending:**
  - Utilizes volume density rather than raw pixel space only
  - Blend images in NeRF's latent representation space

Comparison with Baselines



(a) 2D blending (b) 2D blending with 3D-aware alignment (c) Proposed method (Ours)

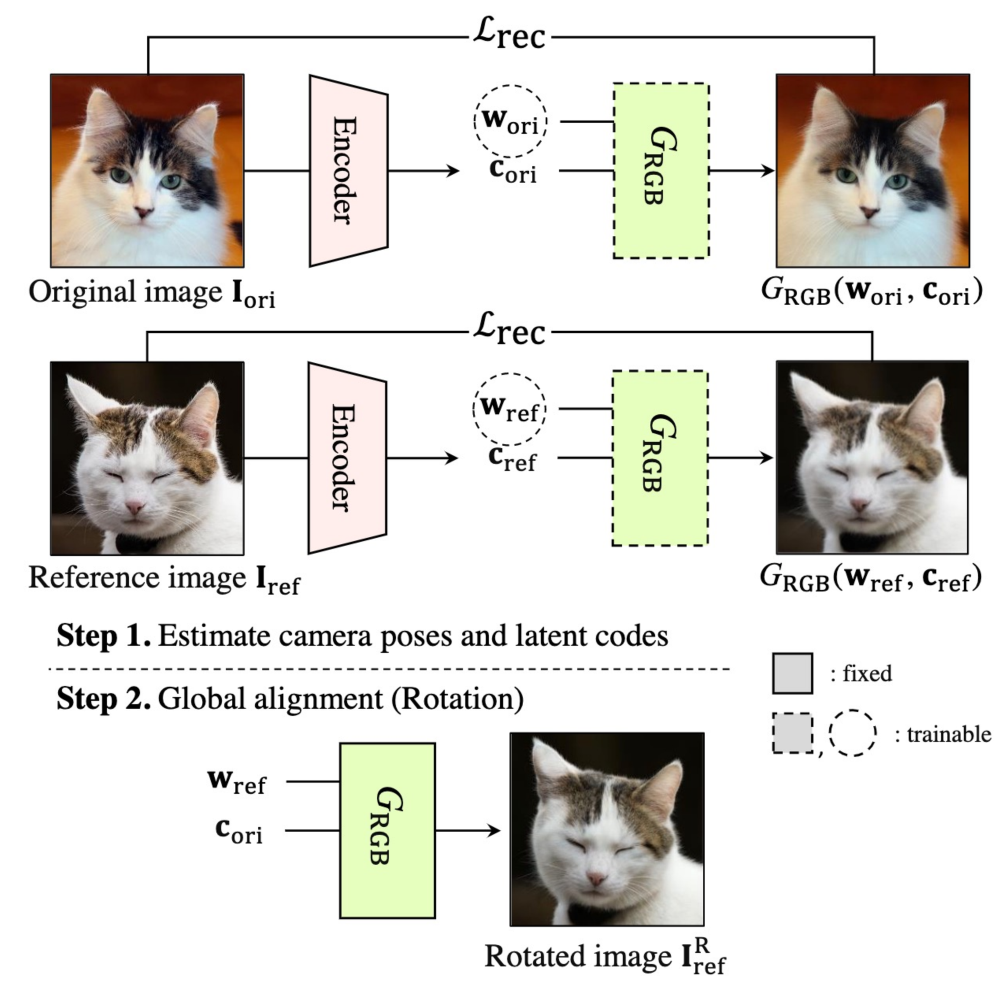
Red lines denote target blending parts.

(a) **2D blending**: 2D blending methods compose two images without any 3D-aware alignment.

(b) **2D blending with 3D-aware alignment**: To address misalignment, we apply our 3D-aware alignment method to existing 2D blending methods.

(c) **Proposed method**: We propose 3D-aware blending after applying our 3D-aware alignment. Note that all methods do not use 3D labels or 3D morphable models.

3D-aware alignment



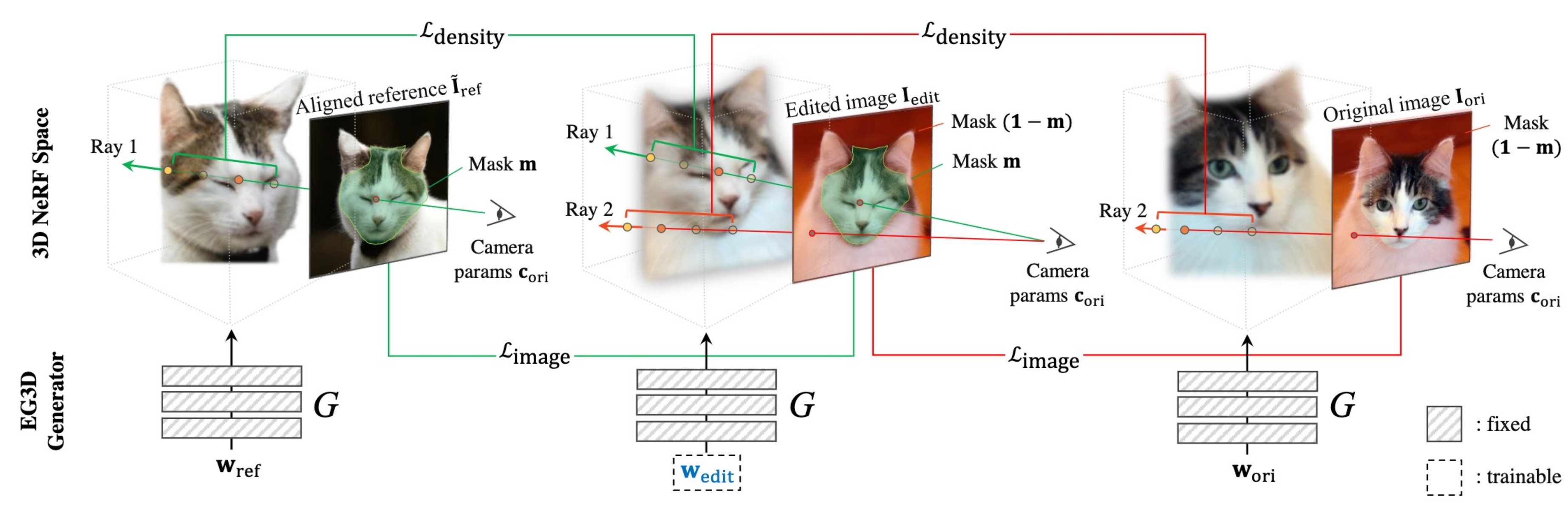
We first use a CNN encoder to infer the camera pose of each input image.

Step 1. Given the camera pose $\mathbf{c}$, we estimate the latent code $\mathbf{w}$ for each input using a reconstruction loss $\mathcal{L}_{\text{rec}}$.

Step 2. Given the estimated camera pose $\mathbf{c}_{\text{ori}}$ and latent code $\mathbf{w}_{\text{ref}}$, we align the reference image to match the pose of the original image.

$$\mathbf{I}^{\text{R}}_{\text{ref}} = G_{\text{RGB}}(\mathbf{w}_{\text{ref}}, \mathbf{c}_{\text{ori}})$$

3D-aware blending



We aim to find the best latent code $\mathbf{w}_{\text{edit}}$ to synthesize a seamless and natural output. To achieve this goal, we exploit both 2D pixel constraints (RGB value) and 3D geometric constraints (volume density). With the proposed image-blending and density-blending losses, we optimize the latent code $\mathbf{w}_{\text{edit}}$.

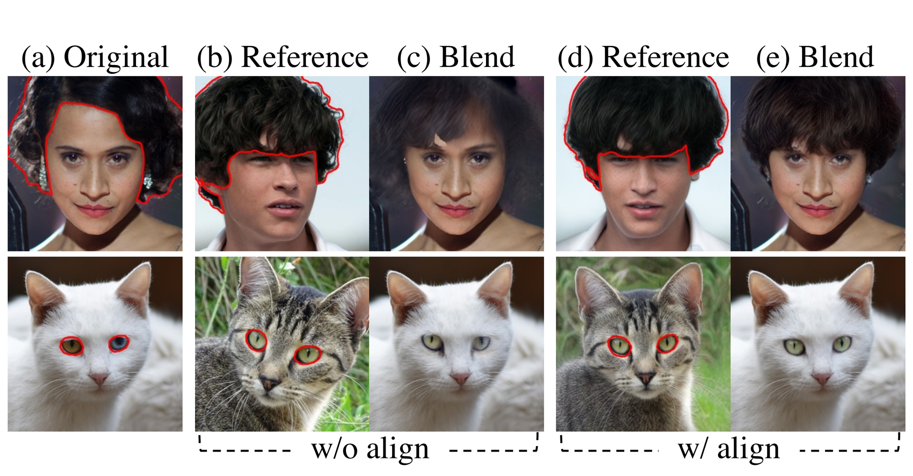
Image blending loss

$$\begin{aligned}\mathcal{L}_{\text{image}} = &\ \|(\mathbf{1}-\mathbf{m}) \circ \mathbf{I}_{\text{edit}} - (\mathbf{1}-\mathbf{m}) \circ \mathbf{I}_{\text{ori}}\|_1 \\ &+ \lambda_1 \mathcal{L}_{\text{LPIPS}}((\mathbf{1}-\mathbf{m}) \circ \mathbf{I}_{\text{edit}}, (\mathbf{1}-\mathbf{m}) \circ \mathbf{I}_{\text{ori}}) \\ &+ \lambda_2 \mathcal{L}_{\text{LPIPS}}(\mathbf{m} \circ \mathbf{I}_{\text{edit}}, \mathbf{m} \circ \mathbf{I}_{\text{ref}}),\end{aligned}$$

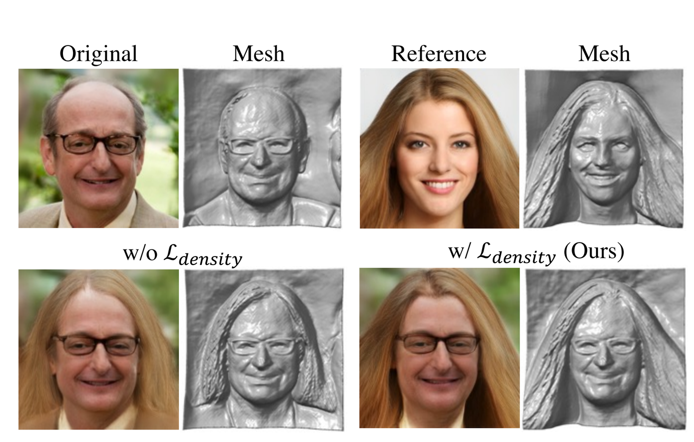
Density blending loss

$$\begin{aligned}\mathcal{L}_{\text{density}} = &\sum_{\boldsymbol{r} \in \mathcal{R}_{\text{ref}}} \sum_{\boldsymbol{x} \in \boldsymbol{r}} \|G_\sigma(\mathbf{w}_{\text{edit}}; \boldsymbol{x}) - G_\sigma(\mathbf{w}_{\text{ref}}; \boldsymbol{x})\|_1 \\ &+ \sum_{\boldsymbol{r} \in \mathcal{R}_{\text{ori}}} \sum_{\boldsymbol{x} \in \boldsymbol{r}} \|G_\sigma(\mathbf{w}_{\text{edit}}; \boldsymbol{x}) - G_\sigma(\mathbf{w}_{\text{ori}}; \boldsymbol{x})\|_1.\end{aligned}$$

Ablation studies



Exp 1. 3D-aware alignment



Exp 2. 3D-aware blending

Quantitative results

Method	w/o align (baseline only) KID ↓	LPIPS$_m$ ↓	mL_2 ↓	w/ 3D-aware align KID ↓	LPIPS$_m$ ↓	mL_2 ↓
Poisson Blending [72]	<u>0.006</u>	0.4203	0.0069	<u>0.005</u>	0.2355	0.0051
Latent Composition [11]	0.012	0.4735	0.0388	0.012	0.4487	0.0321
StyleGAN3 $\mathcal{W}$ [45]	0.016	0.4379	0.0353	0.017	0.3921	0.0307
StyleGAN3 $\mathcal{W}+$ [45]	0.025	0.4634	0.0462	0.023	0.4086	0.0391
StyleMapGAN (32×32) [50]	0.007	0.3792	0.0118	0.006	<u>0.1989</u>	0.0045
SDEdit [64]	0.011	0.3857	0.0076	0.008	0.3427	**0.0003**
Ours	0.013	<u>0.2046</u>	<u>0.0050</u>	0.013	0.2046	0.0050
Ours + Poisson Blending	**0.002**	**0.1883**	**0.0007**	**0.002**	**0.1883**	<u>0.0007</u>

Blending results in various datasets

EG3D backbone : CelebA-HQ, AFHQ-Cat, ShapeNet-Car

StyleSDF backbone: FFHQ