Supplementary Materials for *TexFusion*: Synthesizing 3D Textures with Text-Guided Image Diffusion Models

Tianshi Cao [1,2,3] Karsten Kreis [1] Sanja Fidler[1,2,3] Nicholas Sharp[1,*] Kangxue Yin[1,*]

[1] NVIDIA, [2] University of Toronto, [3] Vector Institute

{tianshic, kkreis, sfidler, nsharp, kangxuey}@nvidia.com

1. Algorithm Details

Algorithm We present a full itinerary for the Sequential Interlaced Multiview Sampler in Algorithm 1 and a simplified block diagram in Fig.1. The symbol $\boldsymbol{I}$ denotes a matrix/tensor of ones of the appropriate size. We elaborate on our choices for the hyperparameters in the following paragraphs. For all other hyperparameters not explicitly specified below (such as the values of α_i), we follow the default settings provided in Stable Diffusion 2's public repository[1].

Adapting DDIM schedule We use DDIM [4] as the basis for configuring our sampler. We use the accelerated denoising process with 50 time steps, uniformly spaced. We truncate the time-step range to $(300, 1000)$ to prevent the network from focusing too much on artifacts introduced when rendering the latent texture map into latent images. At the last denoising step $i = 1$, we perform sequential aggregation at the setting of $t_{i-1} = 300$, but additionally compute $\mathbf{x}_0$ predictions $\hat{\mathbf{x}}_{0,n} = \frac{\mathbf{x}_{i,n} - \sqrt{1-\alpha_i}\epsilon_\theta^{t_i}(\mathbf{x}_{i,n})}{\sqrt{\alpha_i}}$ as final outputs. Following DDIM, we parameterize the noise scale of the DDIM process as $\sigma_i = \eta\sqrt{(1-\alpha_{i-1})/(1-\alpha_i)}\sqrt{1-\alpha_i/\alpha_{i-1}}$. To maximize the consistency of updates produced in each viewpoint, we further introduce a temperature parameter $0 \leq \tau \leq 1$ which scales the noise term. Choosing $\tau < 1$ reduces the variance of the posterior $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$ without effecting its expectation. In the results presented in the manuscript, we use $\eta = 1, \tau = 0.5$ in the coarse stage, and $\eta = 1, \tau = 0$ in the high-resolution refinement stage, which we find to be the most robust configuration.

Classifier-free guidance We use classifier-free guidance to control the alignment of texture to both depth and text. Specifically, we apply classifier-free guidance to both depth and text according to this formula: $\epsilon'^{t_i}_\theta(\mathbf{x}_{i,n}; d_n, \text{text}) = (1 - w_{joint})\epsilon^{t_i}_\theta(\mathbf{x}_{i,n}) + w_{joint}\epsilon^{t_i}_\theta(\mathbf{x}_{i,n}; d_n, \text{text})$. We set $w_{joint} = 5$, and use $\epsilon'^{t_i}_\theta(\mathbf{x}_{i,n})$ in place of $\epsilon^{t_i}_\theta(\mathbf{x}_{i,n})$ in all experiments. We note that this formula is different from the different from that used in SD2/w-depth, which only applies classifier-free guidance to the text prompt by including depth conditioning in both terms on the RHS of the equation.

[1] https://github.com/Stability-AI/stablediffusion

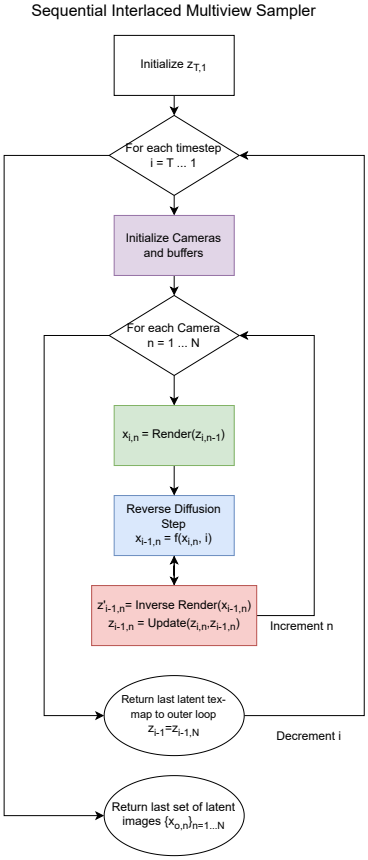


Figure 1: Simplified block diagram of SIMS.

For human heads and bodies, we find that stronger text guidance is helpful for stylization. Thus, we add a text-condition only term as follows: $\epsilon'^{t_i}_\theta(\mathbf{x}_{i,n}; d_n, \text{text}) = (1 - w_{joint} - w_{text})\epsilon^{t_i}_\theta(\mathbf{x}_{i,n}) + w_{joint}\epsilon^{t_i}_\theta(\mathbf{x}_{i,n}; d_n, \text{text}) + w_{text}\epsilon^{t_i}_\theta(\mathbf{x}_{i,n}; \text{text})$. We set $w_{text} = 3$ for these prompts.

Additional geometry processing We align objects with meaningful "front" to face the $+x$ direction, and ensure all objects are placed with $+y$ as "up". Following [2], we augment prompts with "{prompt}, front/side/rear/top-

view" based on the location of the camera to the nearest exact direction; "top-view" is used when the elevation of the camera is above $60°$. Perspective cameras are placed facing the origin at a fixed distance of 1.5 from the origin, and adjust the FOV to fit the object within the image. For most objects, we find that a set of nine cameras - all looking at the origin, eight spaced uniformly surrounding the object (azimuth from $0°$ to $315°$ spaced $45°$ apart, at an elevation of $30°$), and one camera looking down the $-y$ direction - to work reasonable well for objects with reasonable aspect ratios and few occlusions.

In the first round of SIMS sampling, we apply $10°$ random jitters to the elevation and azimuth of each camera, and re-sample each camera for a total of 18 cameras to ensure surface coverage. In the second round, we do not apply jittering and use the fixed set of nine cameras. For human characters, the default set of nine cameras does not adequately cover the entire surface due to occlusions. We instead use 3 sets of 8 cameras: each set is placed radially looking at the y axis (azimuth from $0°$ to $315°$ spaced $45°$ apart), and a different offset is applied to the cameras' y position depending on the set $(0.3, 0.0, -0.3$ respectively). This forms a cylinder of cameras looking at the y axis, and adequately covers all surfaces on the human character geometry.

2. Additional Results

2.1. Qualitative Results

We provide in the supplementary video multi-view renderings of all examples we show in the main paper. Further, in this document, we provide additional results of our method in Fig. 4 and Fig. 5, and comparison to two additional baselines in Fig. 2 as described in Sec. 2.2.

2.2. Additional Baselines

We include two additional methods for qualitative comparison. First is stable-dreamfusion [5], a community-implemented version of Dreamfusion [2] that replaces the proprietary Imagen diffusion model with Stable Diffusion 1.4. Although stable-dreamfusion is a text-to-3D method, not text-to-texture, we include it in our experiments because it is a recently released method and it illustrates the difficulty of jointly synthesizing geometry and texture. We use the default hyperparameters provided in this repository[2], which performs SDS optimization for $10{,}000$ iterations, with a classifier free guidance weight of 100. The second baseline method is the latent-painter variant of latent-nerf [1], for synthesizing textures on an input mesh. Latent-painter performs the same task as us, namely text and geometry-conditioned texture generation, but it does so using the SDS optimization, akin to [2]. We include this method as it was recently the state-of-the-art in texture synthesis with 2D image priors. We use the default hyperparameters provided with this repository[3], which performs

Figure 2: Visual comparison of texture generated by Stable DreamFusion (left) [5], Latent-Painter (middle) [1], and our TexFusion (right). Prompts are cherry picked for those where Stable DreamFusion successfully converged to a reasonable geometry.

[2]https://github.com/ashawkey/stable-dreamfusion

[3]https://github.com/eladrich/latent-nerf

Algorithm 1 Sequential Interlaced Multiview Sampler

Input: mesh $\mathcal{M}$, cameras $\{C_1, \dots, C_N\}$
Parameters: Denoising time schedule $\{t_i\}_{i=T}^{0}$, DDIM noise schedule $\{\sigma_i\}_{i=T}^{0}$, DDIM noise scale η, temperature τ, function for camera jittering $maybe\_apply\_jitter$
$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $i \in \{T \dots 1\}$ **do**
  Init mask $M_i = 0$ of shape (N, H, W)
  Init quality buffer $Q_i = -\infty$ of shape (N, H, W)
  $\mathbf{z}_{i-1,0} = \mathbf{z}_i$
  Apply camera jitter $\{C_{i,1}, \dots, C_{i,N}\} = maybe\_apply\_jitter(\{C_1, \dots, C_N\})$
  Sample forward noise ϵ_i
  for $n \in \{1 \dots N\}$ **do**
    Compute forward diffusion term $\mathbf{z}_{i,n} = M_i \odot \left(\sqrt{\frac{\alpha_{i-1}}{\alpha_i}} \mathbf{z}_{i-1,n-1} + \sqrt{1 - \frac{\alpha_t}{\alpha_{t-1}}} \epsilon_i \right) + (\mathbf{1} - M_i) \odot \mathbf{z}_i$
    Render latent image and compute screen space derivatives $\mathbf{x}'_{i,n}, (\frac{\partial \mathbf{u}}{\partial \mathbf{p}}, \frac{\partial \mathbf{v}}{\partial \mathbf{p}}, \frac{\partial \mathbf{u}}{\partial \mathbf{q}}, \frac{\partial \mathbf{v}}{\partial \mathbf{q}}) = \mathcal{R}(\mathbf{z}_{i,n}; C_{i,n})$
    $J_{i,n} = \left| \frac{\partial \mathbf{u}}{\partial \mathbf{p}} \cdot \frac{\partial \mathbf{v}}{\partial \mathbf{q}} - \frac{\partial \mathbf{u}}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{v}}{\partial \mathbf{p}} \right|$
    Sample $\varepsilon_{i,n} \sim \mathcal{N}(0, \boldsymbol{I})$
    Perform denoising: $\mathbf{x}_{i-1,n} = \sqrt{\alpha_{i-1}} \left(\frac{\mathbf{x}_{i,n} - \sqrt{1-\alpha_i} \epsilon_\theta^{t_i}(\mathbf{x}_{i,n})}{\sqrt{\alpha_i}} \right) + \sqrt{1 - \alpha_{i-1} - \sigma_i^2} \cdot \epsilon_\theta^{t_i}(\mathbf{x}_{i,n}) + \tau \cdot \sigma_i \cdot \varepsilon_{i,n}$
    if $i = 1$ **then**
      $\mathbf{x}_0$ prediction: $\hat{\mathbf{x}}_{0,n} = \frac{\mathbf{x}_{i,n} - \sqrt{1-\alpha_i} \epsilon_\theta^{t_i}(\mathbf{x}_{i,n})}{\sqrt{\alpha_i}}$
    end if
    $\mathbf{z}'_{i-1,n} = \mathcal{R}^{-1}(\mathbf{x}_{i-1,n}; C_{i,n})$
    $Q_{i,n} = \mathcal{R}^{-1}(-J_{i-1,n}; C_{i,n})$
    $M_{i,n} = \mathcal{R}^{-1}(\boldsymbol{I}(\mathbf{x}_{i-1,n}); C_{i,n})$
    Determine update area $U = M_{i,n}(u, v) > 0$, and $Q_{i,n} > Q_i$
    Update pixels $\mathbf{z}_{i-1,n} = U \odot \mathbf{z}'_{i-1,n} + (1 - U) \odot \mathbf{z}_{i-1,n}$
    Update mask and quality buffer $M_i = \max(M_i, M_{i,n})$, $Q_i = \max(Q_i, Q_{i,n})$ (max is applied element-wise)
  end for
  $\mathbf{z}_{i-1} = \mathbf{z}_{i-1,N}$
end for
return $\{\hat{\mathbf{x}}_{0,n}\}_{n=1}^{N}$

5,000 iterations of SDS optimization, also with a classifier free guidance weight of 100.

Results from these two baselines, along with results from TexFusion on the same prompts, can be found in Fig. 2. Stable Dreamfusion failed to converge at all for most prompts in our dataset (*e.g.* Fig. 3), so we selected prompts where Stable DreamFusion did produce reasonable geometry for visualization. This outcome highlights the fragility of optimizing 3D geometry and texture jointly. We find that Latent-Painter often produced over-saturated colors in the texture due to the use of the SDS optimization with high guidance weights. Furthermore, we find significant artifacts in Latent-Painter results that are reminiscent of incorrect UV mapping. This artifact is in fact due to Latent-Painter applying Stable Diffusion's decoder to the latent texture map directly in texture space, thereby creating artifacts at all boundaries of UV islands. Our method does not suffer from the same issue because we apply the decoder to multi-view latent images, making our method agnostic to the underlying UV parameterization.

Figure 3: Example result of Stable-DreamFusion where the geometry did not converge properly. Prompt is "ambulance, white paint with red accents".

2.3. Runtime Comparison

We compare the runtime of TexFusion to baselines running on a workstation with a single NVIDIA RTX A6000

Figure 4: Top: TexFusion + ControlNet in "guess mode"; bottom: TexFusion + ControlNet in "normal mode".

Method	Runtime
stable-dreamfusion	39 min
Latent Painter	22 min
TEXTure (reported in [3])	5 min
TEXTure (ran on our setup)	2.9 min
TexFusion (24 cameras)	6.2 min
TexFusion (9 cameras)	2.2 min

Table 1: Runtime comparison: wall clock time elapsed to synthesize one sample

GPU in Tab. 1. We separately measure the runtime of our method under two different camera configurations (see Appendix Section 1 for details of the camera configuration). We find TexFusion to be an order of magnitude faster than methods that rely on optimizing a neural representation with SDS (17.7x w.r.t stable-dreamfusion and 10x w.r.t. Latent Painter). Our runtime is similar to the concurrent work of TEXTure (2.9 min), whose runtime falls between the 9 camera configuration of our method (2.2 min) and 24 camera configuration of our method (6.2 min). Of the 2.2 min duration, 76 seconds are spent on the first round of SIMS, 22 s on the second round, and 34 s on optimizing the neural color field.

3. Experiment details

3.1. User study details

We conduct a user study using Amazon Mechanical Turk https://www.mturk.com/. We ask each survey participant to look at one pair of texturing results generated by TEXTure and TexFusion according to the same prompt, displayed side-by-side in random left-right order, and answer four questions. For each prompt, we show the survey to 3 participants. We then aggregate the results over all responses. A screenshot of one such survey is shown in Fig. 6.

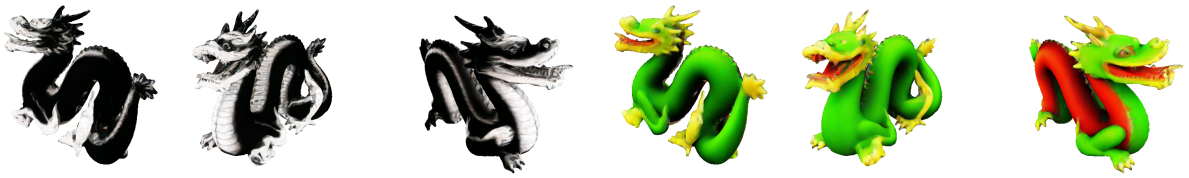
3.2. Dataset description

We collect 35 meshes from various sources. A complete list can be found in Tab. 2 and Tab. 3. Objects from shapenet are selected from ShapeNetCore.v1, obtained under the ShapeNet license[4]. One Human model is obtained from Text2Mesh repository[5]. Objects "house" and "casa" are obtained for free from Turbosquid with permissive licensing. "bunny" and "dragon" are obtained from Stanford 3D scans[6]. "Hermanaubis" and "Provost" are obtained from 3D scans[7], which are shared freely without copyright restrictions. All other objects are obtained under appropriate commercial licenses.

[4] https://shapenet.org/terms
[5] https://github.com/threedle/text2mesh/tree/main/data/source_meshes
[6] http://graphics.stanford.edu/data/3Dscanrep/
[7] https://threedscans.com/

Object	Source	Description	Prompts
1a64bf1e658652ddb11647ffa4306609	shapenet	SUV	“lamborghini urus” “pink porsche cayenne” “white mercedes benz SUV” “green ambulance with red cross”
1a7b9697be903334b99755e16c4a9d21	shapenet	coupe	“silver porsche 911” “blue bmw m5 with white stripes” “red ferrari with orange headlights” “beautiful yellow sports car”
1a48d03a977a6f0aeda0253452893d75	shapenet	pickup truck	“black pickup truck” “old toyota pickup truck” “red pickup truck with black trunk”
133c16fc6ca7d77676bb31db0358e9c6	shapenet	luggage box	“blue luggage box” “black luggage with a yellow smiley face”
1b9ef45fefefa35ed13f430b2941481	shapenet	handbag	“white handbag” “turquoise blue handbag” “black handbag with gold trims”
54cd45b275f551b276bb31db0358e9c6	shapenet	backpack	“red backpack” “camper bag, camouflage” “black backpack with red accents”
e49f6ae8fa76e90a285e5a1f74237618	shapenet	handbag	“crocodile skin handbag” “blue handbag with silver trims” “linen fabric handbag”
2c6815654a9d4c2aa3f600c356573d21	shapenet	lounge chair	“leather lounge chair” “red velvet lounge chair”
2fa970b5c40fbfb95117ae083a7e54ea	shapenet	two-seat sofa	“soft pearl fabric sofa” “modern building in the shape of a sofa”
5bfee410a492af4f65ba78ad9601cf1b	shapenet	bar stool	“yellow plastic stool with white seat” “silver metallic stool”
97cd4ed02e022ce7174150bd56e389a8	shapenet	dinning chair	“wooden dinning chair with leather seat” “cast iron dinning chair”
5b04b836924fe955dab8f5f5224d1d8a	shapenet	bus	“yellow school bus”
7fc729def80e5ef696a0b8543dac6097	shapenet	taxi sedan	“new york taxi, yellow cab” “taxi from tokyo, black toyota crown”
85a8ee0ef94161b049d69f6eaea5d368	shapenet	van	“green ambulance with red cross” “ambulance, white paint with red accents” “pink van with blue top”
a3d77c6b58ea6e75e4b68d3b17c43658	shapenet	beetle	“old and rusty volkswagon beetle” “red volkswagon beetle, cartoon style”
b4a86e6b096bb93eb7727d322e44e79b	shapenet	pickup truck	“classic red farm truck” “farm truck from cars movie, brown, rusty”
fc86bf465674ec8b7c3c6f82a395b347	shapenet	sports car	“batmobile” “blue bugatti chiron”
person	Text2Mesh	Human model	“white humanoid robot, movie poster, main character of a science fiction movie” “comic book superhero, red body suit” “white humanoid robot, movie poster, villain character of a science fiction movie”

Table 2: Description of all geometries used in our dataset, (continued in Tab. 3)

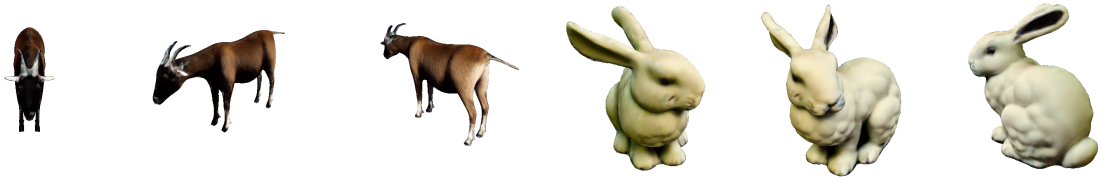


“black and white dragon in chinese ink art style”

“cartoon dragon, red and green”

“blonde girl with green eyes, hair in a tied bun, anime illustration, portrait”

“Portrait of a humanoid robot, futuristic, science fiction”



“brown mountain goat”

“white bunny”

“portrait of greek-egyptian deity hermanubis, lapis skin and gold clothing”

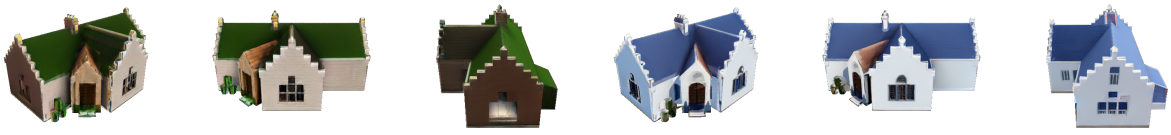
“sandstone statue of hermanubis”

“white fox”

“cartoon fox”

“nunn in a black dress”

“nunn in a white dress, black headscarf”



“minecraft house, bricks, rock, grass, stone”

“colonial style house, white walls, blue ceiling”

Figure 5: Gallery of meshes textured by TexFusion .

Object	Source	Description	Prompts
rp_alvin_rigged_003_MAYA	Renderpeople	Human model	"person wearing black shirt and white pants" "person wearing white t-shirt with a peace sign"
rp_alexandra_rigged_004_MAYA	Renderpeople	Human model	"person in red sweater, blue jeans" "person in white sweater with a red logo, yoga pants"
rp_adanna_rigged_007_MAYA	Renderpeople	Human model	"nunn in a black dress" "nunn in a white dress, black headscarf"
rp_aaron_rigged_001_MAYA	Renderpeople	Human model	"railroad worker wearing high-vis vest" "biker wearing red jacket and black pants"
Age49-LoganWade	Tripleganger	Human head	"oil painting of a bald, middle aged banker with pointed moustache" "moai stone statue with green moss on top" "portrait photo of abraham lincoln, full color"
Age26-AngelicaCollins	Tripleganger	Human head	"Portrait of a humanoid robot, futuristic, science fiction" "blonde girl with green eyes, hair in tied a bun, anime illustration, portrait" "blonde girl with green eyes, hair in tied a bun, DSLR portrait photo"
house	Turbosquid	Medieval house	"medieval celtic House, stone bricks, wooden roof" "minecraft house, bricks, rock, grass, stone" "colonial style house, white walls, blue ceiling"
casa	Turbosquid	house in the sea	"white house by the dock, green ceiling, cartoon style" "minecraft house, bricks, rock, grass, stone" "white house by the dock, green ceiling, impressionist painting"
1073771	Turbosquid	rabbit	"brown rabbit" "purple rabbit" "tiger with yellow and black stripes"
1106184	Turbosquid	cartoon dog	"cartoon dog" "lion dance, red and green" "brown bull dog"
1117733	Turbosquid	goat	"brown mountain goat" "black goat with white hoofs" "milk cow"
1281334	Turbosquid	cartoon cow	"cartoon milk cow" "giant panda"
1367642	Turbosquid	cartoon fox	"cartoon fox" "brown wienner dog" "white fox"
bunny	Stanford 3D Scans	bunny	"white bunny"
dragon	Stanford 3D Scans	dragon	"black and white dragon in chinese ink art style" "cartoon dragon, red and green"
Hermanubis	3D scans	statue	"sandstone statue of hermanubis" "portrait of greek-egyptian deity hermanubis, lapis skin and gold clothing"
Provost	3D scans	statue	"portrait of Provost, oil paint" "marble statue of Provost"

Table 3: Description of all geometries used in our dataset continued.

References

[1] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 2

[2] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1, 2

[3] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 4

[4] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 1

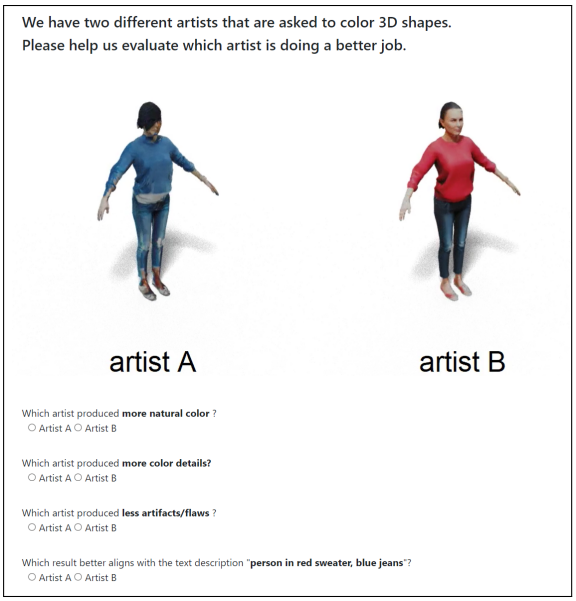


Figure 6: Screenshot of example user study screen

[5] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion. 2