Best Segmentation Buddies for Image-Shape Correspondence

Supplementary Material

The following sections provide more information regarding our image-shape correspondence method. Sec. A refers to the BSB algorithm summary. Sec. B presents additional results and experiments we conducted with our method. Secs. C and D discuss our quantitative evaluation on the PartNet dataset and the perceptual user study, respectively. In Sec. E, we detail ablation test results for verifying our design choices. Finally, Sec. F elaborates on implementation details for the vision model distillation to the 3D shape and the comparison settings with previous work.

A. The Best Segmentation Buddies Algorithm

For clarity, we summarize the BSB matching algorithm in Algorithm 1. As explained in the paper, the algorithm includes finding candidate vertices via feature similarity between the clicked image pixel and the mesh vertices, and selecting the vertex whose most similar pixel falls within the part mask and produces the most consistent segmentation with that mask.

B. Additional Results

Robustness to shape texture. To examine the tolerance of our method to shape texture, we used 3D Paintbrush (without localization) [10] to paint untextured meshes, and ran our BSB matching scheme, where we distilled 3D vision features from renderings of the textured shape. We found out that our method is robust to the shape's texture, as exemplified in Fig. 12, where the object in the image has a different appearance.

Interestingly, we did not observe a clear advantage in using textured shapes. Our BSB mechanism finds a matching vertex in a corresponding semantic region of the shape. Thus, it handles feature discrepancy between the image and the shape and produces robust correspondences when the modalities differ in their texture, and even when the shape is untextured.

Interactive correspondence. Since both the 2D and 3D segmentation models are interactive, our BSB mechanism enables *interactive* cross-modality cross-domain correspondence, as Fig. 13 demonstrates. In this experiment, for each clicked pixel, we found the best segmentation buddy vertex. Then, we used the corresponding pixel-vertex pairs successively for the image and the shape, respectively, yielding the interactive edit of the image-shape correspondence.

Different object poses. Our method can match a region in an image to a 3D shape part when the objects have different poses, as presented in Fig. 6 for the rigid hammer shape.

Figure 12. **Texture robustness.** BSB matches semantic regions between image and shape despite variations in their appearance and texture.

In Fig. 14, we show another aspect of this property, demonstrating correspondences for the human body - a deformable object in different poses. As seen in the figure, BSB succeeds in matching body parts for extreme pose changes: Spider-Man's head in its famous upside-down pose to an upright A-pose person head; a single hand stand where the right arm is extended to the side to a 3D curved right arm oriented downwards; and the left leg swinging up in a Capoeira cartwheel movement to the downward standing 3D leg. These examples further exemplify the robustness of our correspondences.

Multi-region correspondence. In Fig. 6 in the paper, we have shown multiple region correspondences between the same image and shape. Fig. 15 shows additional such example. In addition to matching an image and a shape of the same object type (the lamp, Fig. 6), we can also find matches when the overall objects are different, but their parts have a similar semantic meaning (the lamp and goblet pair, Fig. 15), or when the object has several similar part instances (the backrest and seat pillows and the armrests in

Figure 13. **Interactive correspondence.** Our BSB matching between pixel clicks and mesh vertices, combined with interactive 2D and 3D segmentation, enables to dynamically update the cross-modality correspondence.

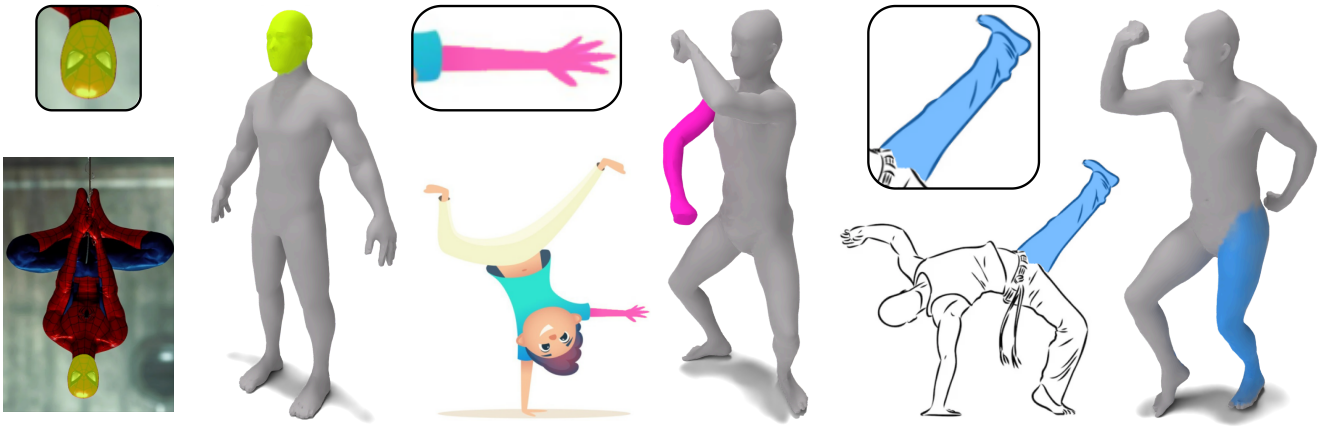

Figure 14. **Matching between differently posed objects.** BSB finds correspondence between the image and shape when the object in each modality differs substantially.

the couch pair, Fig. 15). These results highlight the correspondence specificity of our method.

One view to a 3D part. Our method corresponds a part segment in a 2D image to a complete part in a 3D mesh, as we show in Fig. 16. Surprisingly, even though the image contains only one viewpoint of the underlying object, we are still able to accurately match the entire corresponding region in 3D (see the backrest, seat, and leg in Fig. 16). We hypothesize that this capability can be partially attributed to the fact that we render the mesh from multiple views and lift deep visual features from each of the views onto the shape surface. Thus, a relevant viewpoint of the shape is likely to be present during the distillation process, enabling the matching of a pixel-vertex pair from the corresponding semantic regions.

Figure 15. **Multi-region correspondence.** BSB can match multiple regions between the same image-shape pair, when the modalities depict different objects (left), or distinguishing between similar parts of the objects and matching them correctly (right).

Correspondence stability. Fig. 17 shows the Correspondence results for different clicks within the same image region (the armadillo's carapace). While the clicks are matched to different shape vertices, these vertices fall within the semantically matching region in the standing armadillo shape. Thus, the correspondence between the image and shape is maintained for the different image clicks, demonstrating the stability of our method.

Corresponding different images to the same shape. Fig. 18 presents the correspondence between several images and the same mug shape. The images depict various types of object instances (a tea cup, a themed mug, and a glass), where the geometry and appearance of the 3D mug handle differ significantly between the images. Nonetheless, our method can consistently segment the corresponding part in 3D, suggesting its robustness to the part properties. Additionally, as explained in Sec. 4.3 in the main body, a beneficial outcome of this multi-image to the same shape correspondence is the unsupervised region matching between the images (the cup handles and the hat brims in Fig. 18).

Additional image segmentation interfaces. Our method is not limited to click-based segmentation. It supports alternative ways for obtaining the part region in the image, in addition to point-click, offering further flexibility to the user. One such approach is to utilize the 2D segmentation model [22] with a box input, where the user specifies the top-left and bottom-right coordinates in the image to segment the part mask m_p^{2D} used in our matching scheme. Examples are shown in Fig. 19. Another interface for segmenting the image is text, as we describe next.

Text to 3D segmentation. In the main paper, we used a click-based model for segmenting the image [22]. However, another popular interface nowadays for image segmentation is text. Our method can be easily integrated with this interface, as Fig. 21 demonstrates. In this experiment, we ap-

Figure 16. **A single view to a complete 3D part.** Although each image depicts only one view of the object (left), the entire corresponding part is successfully segmented in 3D (right).

plied language-driven image segmentation by predicting a bounding box for an object part described by text, and segmenting the part within the bounding box [48]. Then, we used that part mask and its centroid as the pixel click with our BSB mechanism to find the corresponding vertex and segment the 3D mesh. This process resulted in text-based 3D segmentation: the image is segmented with text, and the matching 3D part is found by our method.

Different vision model backbones. Recent works have devised new feature-extraction methods from 2D foundation models for image-to-image correspondence [19, 33, 53, 67]. Our image-shape correspondence framework is versatile and can incorporate such techniques. Fig. 20 shows an example where Diffusion Image Features (DIFT) [53] were used for finding best segmentation buddies (instead of DINOv2 [43]). For example, this backbone vision model enables matching the wings for image-mesh airplanes. However, we note that DIFT requires text prompts describing the image and the shape, while DINOv2 does not require such inputs from the user. Thus, in our experiments, we have focused on using DINOv2 as the backbone model for feature correspondence.

We note that this experiment is different than the one described above. Before, we replaced the *segmentation model* to use text instead of a click interface, and employed DINOv2 [43] as the vision model. Here, we changed the *vision model* [53] for extracting pixel and vertex features for comparing their similarity, and used click-based image segmentation [22]. Together, these experiments showcase the versatility of our approach, where the different segmentation and vision models can be utilized for diverse and flexible image-shape correspondence.

Missing corresponding shape part. When the shape is missing a part that exists in the image, our BSB scheme will detect that there is no best segmentation buddy vertex, and no part of the shape will be segmented. See examples in Fig. 22.

Shape-to-image correspondence. As explained in the main paper, BSB can be utilized for 3D to 2D matching, when the image and shape switch roles. Additional such examples appear in Fig. 23.

Limitations. Even though we leverage the same segmentation backbone [22] to perform segmentation in 2D and in 3D (after distillation), there may be a mismatch in the segmentation *granularity* due to the modality gap. Thus, even if our method finds the best segmentation buddy in the right

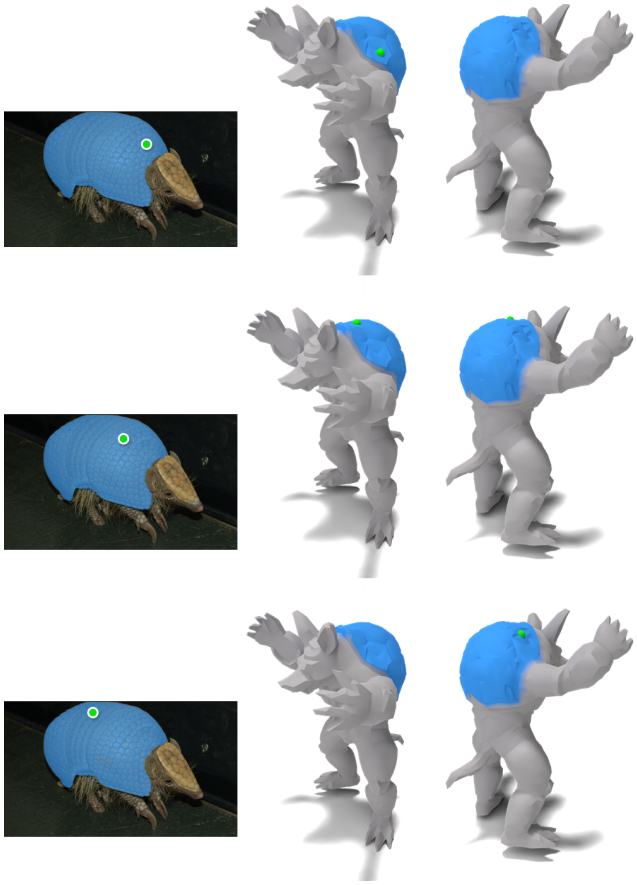

Figure 17. **Correspondence stability.** Our method is robust to the location of the pixel click in the image region (left). Although different pixels are matched to different vertices, they fall within the corresponding semantic 3D part (right), resulting in a stable matching between the image and the shape. The clicked pixel and the matched vertices are visualized with a green dot.

Figure 18. **Different images to the same shape.** BSB accurately matches segmentations from images that contain significant differences in geometry (*e.g.*, the heart-handle on the left) and appearance (*e.g.*, crochet hat on the right) to the same shape.

semantic region of the shape, the 3D and 2D segments may not match. Fig. 25 demonstrates such cases.

For a click on the camel's ear, the 2D model segments the ear. In this case, BSB selects the best segmentation buddy vertex correctly on the corresponding ear of the camel shape. However, the 3D model considers the matched vertex to be a part of the head and segments the entire camel's head. In the case of the horse mesh, while the 2D model segments the entire horse body, the 3D model segments only the belly.

Another limitation arises when one segment in the image corresponds to several parts in the shape or the other way around, as presented in Fig. 26. In the first case, the curved lamp rod in the image matches the two 3D lamp rods. However, since our method selects only one BSB vertex for a pixel click, and the 3D segmentation model [27] separates the rods, only one rod is aligned.

In the second case, the backrest of the hand-shaped chair statue is composed of four "fingers". A click on the pinkie is matched correctly to a vertex on the backrest of the 3D chair. However, while the entire backrest of the mesh is segmented, the 2D model marks only the pinkie region in the image. Thus, in both cases, the resulting image-shape segmentation matching is partial.

Figure 19. **2D region segmentation with a box prompt.** BSB is highly flexible and can be used with a box prompt (the dashed rectangle) for segmenting the part in the image.

C. Quantitative Evaluation

To the best of our knowledge, a dataset with image-shape segment correspondence annotations is nowhere to be found. Thus, we constructed one ourselves from the PartNet dataset [39], as described in Sec. 4.2. PartNet is an annotated 3D part segmentation dataset, containing household objects, such as chairs, hats, etc. To create our cross-modality correspondence dataset, we sampled up to 20 shapes from the dataset's categories (fewer for categories with a smaller number of instances).

From each shape, we selected 10 vertices from a part

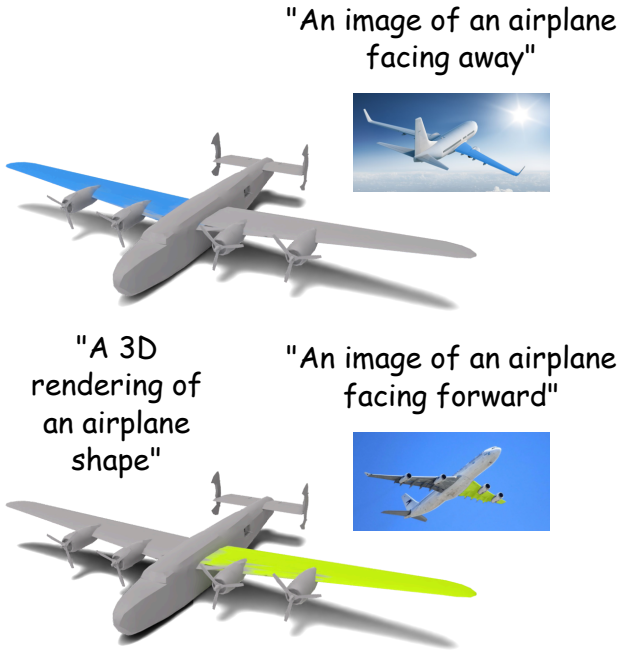

Figure 20. **Backbone model versatility.** BSB can utilize a vision backbone other than DINOv2 [43]. In this case, we lift a diffusion model features [53] to the 3D mesh for finding correspondences. The text prompts used to extract features for the images and the renderings of the shape are indicated next to them.

Figure 21. **Text-based 3D segmentation.** Combining language-driven image segmentation with our method, we achieve 3D segmentation with text. The prompt above the image was used for its segmentation.

at random, and rendered the shape from a set a views, with elevation of $\{-60°, 30°, 0°, 30°, 60°\}$, and azimuth of $\{0°, 30°, ..., 330°\}$, a total of $5 \cdot 12 = 60$ possible views. We randomly selected two of these views in which the vertex was visible. We discarded vertices that were not visible in all views and shape instances for which all 10 vertices were not visible.

This process resulted in 265 shape instances, with a total of 2118 rendered images. For each of these views, we generated an image with ControlNet [68] and projected the 3D click to the 2D view, as explained in Sec. 4.2, and visualized in Fig. 24. To create naturally looking images, we used the prompt "*A picture of a* $\{$*category*$\}$ *in realistic environments*". Accordingly, our dataset contained 2118 image-shape pairs, with a ground-truth 3D region for the pixel click in the generated image.

This dataset was used for the quantitative evaluation of our method compared to the alternative zero-shot image correspondence baselines [3, 53], reported in Tab. 1. Fig. 24 shows example results from the evaluation. As shown in the table and seen in the figure, the competitors fail to compute meaningful correspondences, whereas our method successfully matches the clicked pixel in the generated image to a vertex within the ground-truth region of the 3D shape, showcasing the superiority of our correspondence approach.

As another baseline, we examined the nearest vertex in the feature space as the match to the pixel click. This baseline achieved a success rate of 0.73. We note that since no existing dataset provides ground-truth annotations for image-shape correspondence, we performed the large-scale quantitative evaluation (Tab. 1) on our synthetically generated cross-modality dataset, in which images are generated conditioned on the underlying shape geometry, as demonstrated in Fig. 24.

In this simplified setting, *where the image and shape share the same underlying structure*, nearest-neighbor matching performs well, as correspondence is easier to establish when the object type and structure align. In our target setting, where the image may exhibit substantial structural differences from the shape, BSB significantly outperforms the NN baseline, as evidenced by the perceptual study detailed next in Sec. D.

Figure 22. **Missing shape part.** If a segmented region in the image is missing a matching part in the shape, our method will output an empty 3D segmentation, indicating correctly that correspondence does not exist in this case.

D. Perceptual Study

BSB is not bound to a specific shape category or a given set of parts defined in a dataset. Thus, to evaluate the effectiveness of the flexible correspondences achieved by our method, where ground-truth labels do not exist, we conduct

Figure 23. **Shape-to-image correspondence.** Our method can match a part of the shape to its corresponding semantic segment in the image, even when the image contains a lot of distractors.

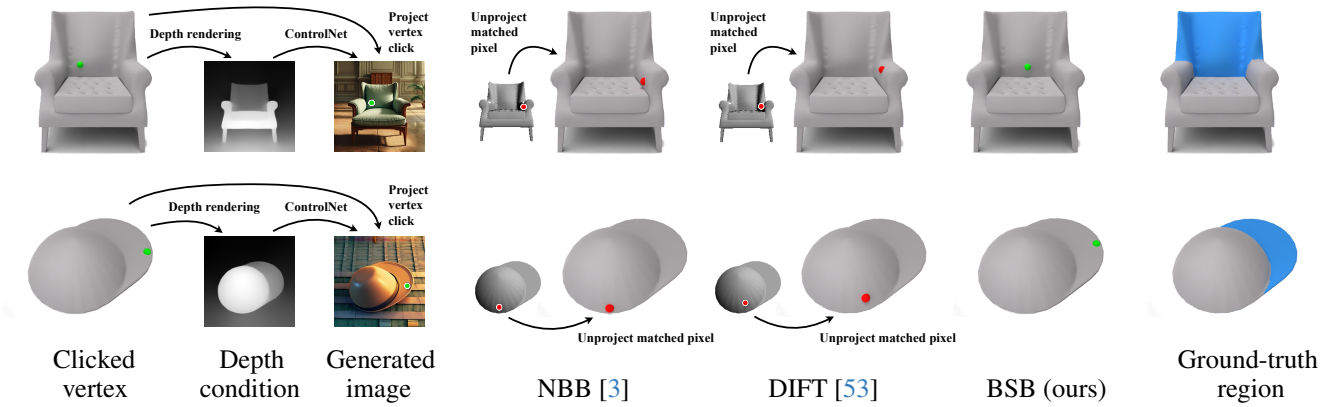

Figure 24. **Correspondence comparison on PartNet.** We show the generation process of the input image and 2D click (first three columns), the matched pixel by NBB and DIFT from the generated image to the rendered image of the shape and its unprojection to 3D (fourth to eighth columns), our matching vertex (ninth column) for the pixel click on the generated image, and the ground-truth shape region (tenth column) from which the vertex was selected (first column). Incorrect and correct correspondences are marked with red and green dots, respectively. While the baselines' correspondences are wrong, our method successfully finds a vertex within the ground-truth shape region.

a perceptual user study. The survey included 31 participants who evaluated 20 image-shape pairs of various objects, including animals, humanoids, and household shapes, where the image and the shape differed in their structure or came from different domains. Figs. 9 and 27 present example image-shape pairs from the perceptual study.

For each pair, we showed the 2D and 3D corresponding segmentations for different methods and asked the participants to rate the effectiveness of the result on a scale of 1 to 5. The score 5 reflects a completely effective matching, where the correct part is segmented in 3D. When another part is segmented, the matching is partially effective, and when a wrong part is selected, or there is an empty segmentation, the correspondence is deemed completely ineffective, reflected by the score 1. We report the effectiveness scores averaged over the participants and the image-shape examples in Tab. 2. Our method is considered substantially more effective than the baselines, highlighting its high fidelity correspondences.

Fig. 27 shows examples for selecting the nearest neighbor vertex for the pixel click in the image. Since there is a mismatch between the vision features of pixels and vertices, in this case, the pixel may be wrongly matched to a vertex in a different semantic region. For instance, the correspondence can be to another similar region in the shape, like the right penguin foot rather than its left one; to a vertex in an adjacent region, like the guitar's head instead of its neck; or to a shape part that looks similar to the image segment but does not match it semantically, as the bunny's tail instead of its ear. In contrast, our BSB considers different vertex candidates and selects the one that maps back to the segmented region in the image, enabling it to mitigate the modality shift and find correct correspondences.

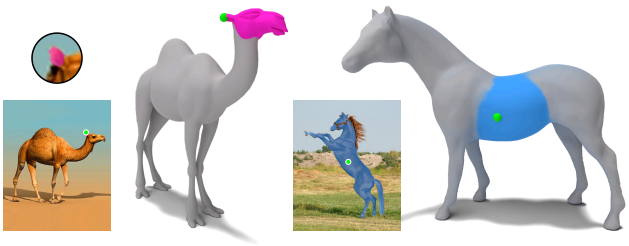

Figure 25. **Segmentation granularity mismatch limitation.** Our method assumes that the 2D and 3D segmentation models exhibit the same granularity for the image and the mesh. However, correct corresponding clicks on the image and the mesh may have different segmentation mask granularity, compromising the resulting segment correspondence between the modalities.

E. Ablation Test

We validate the design choices in our method with ablation experiments. First, we examine the influence of the number

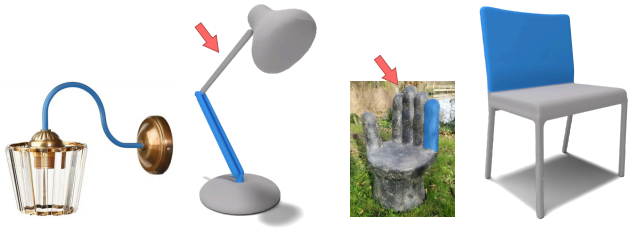

Figure 26. **One-to-many and many-to-one partial matching limitation.** When a single segment in the image corresponds to multiple mesh parts of the same semantic entity (the lamp's body) or vice versa (the chair's backrest), our resulting image-shape correspondence will be partial.

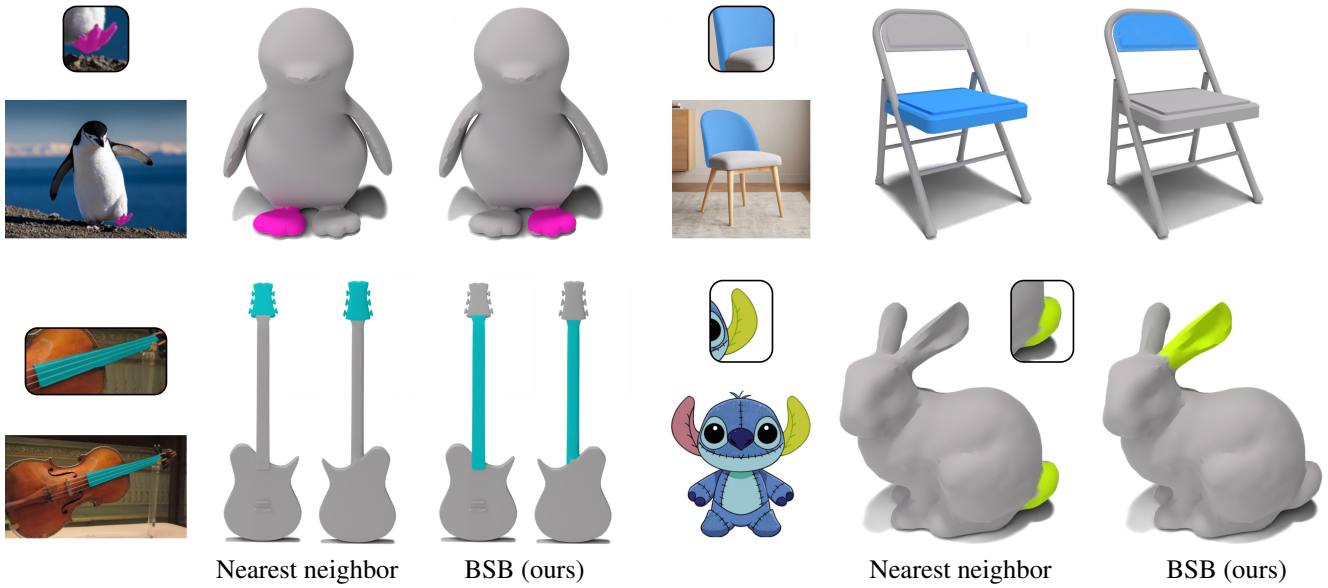

Figure 27. **Nearest neighbor vertex selection.** Selecting the nearest neighbor vertex for a pixel click in the image leads to erroneous correspondences. In contrast, our BSB overcomes the image-shape modality gap and finds correct matches.

Method	NBB	DIFT	NN Baseline	BSB (ours)
Effectiveness ↑	2.75	2.74	3.26	**4.63**

Table 2. **Perceptual user study.** We evaluate the image-shape correspondence effectiveness on a scale of 1 (completely ineffective) to 5 (completely effective). NN Baseline stands for selecting the nearest neighbor vertex in the feature space as the match for the pixel click. Our method is considered much more effective than the competitors.

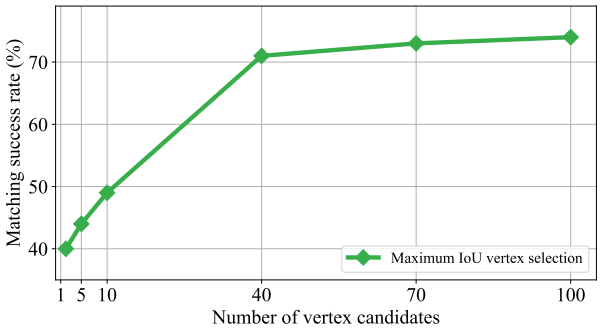

Figure 28. **Different number of vertex candidates.** We evaluate the matching success rate on PartNet for different values of vertex candidates. The performance starts to increase with a higher number of candidates and then saturates.

of vertex candidates k on the correspondence success rate for the image-shape pairs from the quantitative evaluation discussed in Sec. 4.2. Fig. 28 presents the results.

Due to the modality difference, there is a shift in the vision features of pixels and vertices. Thus, when k is too small, a vertex from the corresponding ground-truth region is typically absent from the k candidates of the clicked pixel, resulting in a low correspondence success rate. As k is increased, additional candidates are considered, including ones from the ground-truth region, and a vertex from the correct 3D region is more frequently found by our BSB mechanism, improving the success rate. Finally, the performance saturates at a success rate of 0.74 towards $k = 100$, leading to this configuration of k in our method.

Additionally, we checked the case of randomly selecting a vertex for correspondence out of the $k = 100$ candidates *vs*. choosing the one that maps to a pixel with the highest IoU with the image segmentation region, as proposed in our work. We have found that the former case results in a considerably lower matching success rate of 0.48, validating the effectiveness of the proposed selection procedure for the corresponding vertex.

F. Implementation Details

Vision model distillation. We train a multi-layer perceptron (MLP) to map each mesh vertex to a DINOv2-like feature vector of size $d_{vis} = 1024$. At the network's input, we use the vertex coordinates together with positional encoding of size 2048. The MLP contains 6 layers with 1024 neurons each. Each layer, except the last one, includes ReLU activation and layer normalization. For the last layer, we apply hyperbolic tangent activation without normalization.

For training the MLP, we render images of size 224×244 for 1000 random views of the mesh, where the angles were sampled randomly from the full elevation and azimuth range of $[-180°, 180°]$ and $[0°, 360°)$, respectively. Each

image was encoded by the DINOv2 ViT-L14 model to a tensor of size $w' \times h' \times d_{vis} = 16 \times 16 \times 1024$, and then interpolated to a spatial size of 64×64. Then, we rendered the predicted features for the mesh using a differentiable renderer into a tensor of size $64 \times 64 \times 1024$ and compared them to the reference DINOv2 features using a mean squared error loss. The network was trained with an ADAM optimizer for 3 epochs with a learning rate of 0.001. The entire distillation process, including rendering mesh views, encoding them, and training the network, takes only 3.5 minutes.

Compared methods. For the compared baselines [3, 53], we used the publicly available code bases released by the authors, with their recommended configuration.

Algorithm 1 Best Segmentation Buddies (BSB) matching

Input: Image $\mathcal{I}$ of size (w, h), clicked pixel p, mesh $\mathcal{M}$ with vertices $\mathcal{V}$, 2D segmentation model $\mathscr{F}^{2D}_{seg}$, 2D vision model $\mathscr{F}^{2D}_{vis}$, distilled 3D MLP $\mathscr{F}^{3D}_{vis}$, number of vertex candidates k.

Output: Best segmentation buddy vertex v_p (or `None`).

Preprocess: distill $\mathscr{F}^{2D}_{vis}$ into $\mathscr{F}^{3D}_{vis}$ from multi-view renders of mesh $\mathcal{M}$.
Feature extraction:
$(M^{2D}_o, M^{2D}_p) \leftarrow \mathscr{F}^{2D}_{seg}(\mathcal{I}, p)$
$F'^{\mathcal{I}}_{vis} \leftarrow \mathscr{F}^{2D}_{vis}(\mathcal{I})$
$F^{\mathcal{I}}_{vis} \leftarrow \text{interpolate}(F'^{\mathcal{I}}_{vis}, (w, h))$
$F^{\mathcal{V}}_{vis} \leftarrow \mathscr{F}^{3D}_{vis}(\mathcal{V})$
Best segmentation buddy search:
$s_{pv} \leftarrow \text{cossim}(F^{\mathcal{I}}_{vis}[p], F^{\mathcal{V}}_{vis}[v])$ for all $v \in \mathcal{V}$
$\mathcal{C} \leftarrow$ top-k vertices by s_{pv}
$maxIoU \leftarrow 0$, $v_p \leftarrow$ `None`
for all $v' \in \mathcal{C}$ **do**
  $q' \leftarrow \text{argmax}_{q \in M^{2D}_o} \text{cossim}(F^{\mathcal{V}}_{vis}[v'], F^{\mathcal{I}}_{vis}[q])$
  if $q' \notin M^{2D}_p$ **then**
    continue
  end if
  $M^{2D}_{q'} \leftarrow \mathscr{F}^{2D}_{seg}(\mathcal{I}, q')$
  $iou \leftarrow \text{IoU}(M^{2D}_p, M^{2D}_{q'})$
  if $iou > maxIoU$ **then**
    $maxIoU \leftarrow iou$
    $v_p \leftarrow v'$
  end if
end for
if $v_p =$ `None` **then**
  return `None`
else
  return v_p
end if