DOI: 10.1111/cgf.14197

COMPUTER GRAPHICS *forum*
Volume 40 (2021), number 1 pp. 410–423

SketchZooms: Deep Multi-view Descriptors for Matching Line Drawings

Pablo Navarro,[1,2,3] J. Ignacio Orlando,[3,4] Claudio Delrieux[3,5] and Emmanuel Iarussi[3,6]

[1]Instituto Patagónico de Ciencias Sociales y Humanas, CENPAT, Puerto Madryn, Argentina
[2]Departamento de Informática, Universidad Nacional de la Patagonia San Juan Bosco, Trelew, Argentina
[3]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
[4]Instituto Pladema, UNICEN, Tandil, Argentina
[5]Departamento de Ing. Eléctrica y Computadoras, Universidad Nacional del Sur (UNS), Bahia Blanca, Argentina
[6]Universidad Tecnológica Nacional (UTN FRBA), Buenos Aires, Argentina

Abstract

Finding point-wise correspondences between images is a long-standing problem in image analysis. This becomes particularly challenging for sketch images, due to the varying nature of human drawing style, projection distortions and viewport changes. In this paper, we present the first attempt to obtain a learned descriptor for dense registration in line drawings. Based on recent deep learning techniques for corresponding photographs, we designed descriptors to locally match image pairs where the object of interest belongs to the same semantic category, yet still differ drastically in shape, form, and projection angle. To this end, we have specifically crafted a data set of synthetic sketches using non-photorealistic rendering over a large collection of part-based registered 3D models. After training, a neural network generates descriptors for every pixel in an input image, which are shown togeneralize correctly in unseen sketches hand-drawn by humans. We evaluate our method against a baseline of correspondences data collected from expert designers, in addition to comparisons with other descriptors that have been proven effective in sketches. Code, data and further resources will be publicly released by the time of publication.

Keywords: image and video processing, 2D morphing, image and video processing, image databases, image and video processing

ACM CSS: • Computing methodologies → Neural networks; Image processing

1. Introduction

Humans excel at perceiving 3D objects from line drawings [Her20]. Therefore, freehand sketches are still the preferred way for artists and designers to express and communicate shape without requiring to construct the intended object. Unlike humans, computers struggle to interpret a 2D sketch as a highly condensed abstraction of our 3D world. For instance, the straightforward task of finding correspondences between a pair of images or an image and a 3D model has been an important problem in Computer Graphics and Vision for decades. In comparison with photographs, dealing with sketches is even more challenging [ADN*17], since line drawings lack key shape cues like shading and texture, projections are imprecise, and shapes are often composed by several sketchy lines (Figure 1). Consequently, when a target object is viewed from different angles, traditional image descriptors fail to map similar points close together in the descriptor space. Furthermore, recent studies show that even advanced deep networks lack the ability to generalize to sketches when originally trained to perform perceptual tasks over photo collections [LOVH19].

To date, finding local sketch correspondences with deep learning techniques is an unexplored research topic. This is likely because learning meaningful and consistent features using such high capacity models requires a large dataset of complex line drawings, paired semantically at a dense, pixel-wise level. To overcome this difficulty, our key contribution is a vast collection of synthetic sketches, distributed in several semantic categories. This massive dataset serves to compute local sketch descriptors that can deal with significant image changes. In our setup, a query point is represented by a set of 2D zooms captured from the point's immediate neighbourhood, resulting in multiple zoomed versions of the

Figure 1: *Unlike photographs, typical design sketches lack shading, texture and lines are often rough and incomplete.*

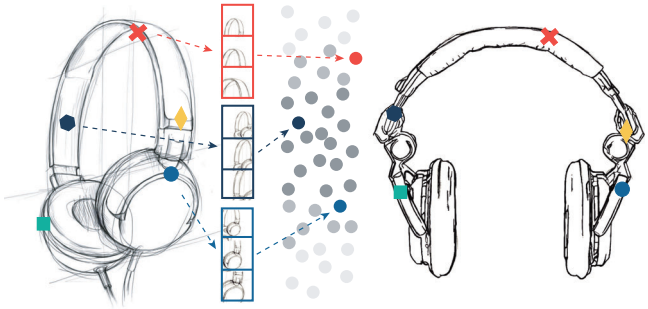

Figure 2: *The multi-view neural network embeds similar points on sketches close to one another in descriptor space despite the significant changes in viewport and shape. By training on a dataset of part-based registered 3D models rendered as sketches, SketchZooms is able to generalize to different rendering styles, incorporating the semantics from an object category.*

corresponding point. The main goal is to capture the domain semantics and object part characteristics despite the heterogeneous nature of hand-drawn images (see Figure 2). Our hypothesis is that learning from such a large database may result in a general model that overcomes the covariate shift between artificial and real sketches.

Although the available literature provides descriptors which are robust to shape variation and affine distortions, to the best of our knowledge this is the first attempt to cope with part semantics and 3D viewport changes in sketches. To this avail, we evaluate and compare SketchZooms against well-known state-of-the-art techniques extensively used in line drawings' applications. Furthermore, the generalization ability of our framework is assessed by evaluating the proposed approach using the OpenSketch image collection, [GSH*19] rendered by designers in different styles and viewports. Taking advantage of co-registered images in OpenSketch, without any prior fine-tuning, we compute quantitative metrics along with the qualitative results (Section 5). Experiments show that this approach is able to deal with significant changes in style, shape and viewport, generalizing well to non-synthetic inputs. Finally, we demonstrate the usefulness of our descriptors for graphics applications such as sketch-based shape retrieval and image morphing.

In summary, our contributions are:

- The first approach applying deep learning to the problem of finding local image correspondences between design sketches.
- A massive co-registered synthetic line drawing dataset rendered from 3D shapes, which allows our trained models to generalize to designer sketches, even from unseen object categories.
- A comprehensive evaluation and comparison with respect to other methods commonly applied to find correspondences in line drawings, including a perceptual study against human-established matchings.

2. Related Work

Finding image descriptors that effectively represent image data is a classic problem in Computer Graphics and Vision. A comprehensive summary of the relevant literature is out of the scope of this paper. Instead, we focus on descriptors involving line drawings, either for registration or retrieval tasks on images and 3D models. We avoid general natural image descriptors like SIFT Low99] which has been shown ineffective to cope with sparse stroke orientations in sketches ERB*12]. We briefly classify them into two main groups: *hand-crafted* and *learned descriptors*.

Hand-crafted descriptors consist on applying custom transformations over some input data in order to obtain a suitable global or local representation. Many applications working with raster input employ pixel-based descriptors. For instance, ShapeContext BMP02] is a well known descriptor that captures the point distribution on a given neighbourhood, which was proven effective for corresponding feature points in sketches [CCT*09, IBT13]. Combined with cycle consistency methods like FlowWeb [ZJLYE15], some authors boosted ShapeContext performance and benefited from the availability of multiple similar sketches [ADN*17]. In the context of vector graphics, several authors proposed to quantify stroke similarity in order to generate in-between frames for character animation [WNS*10, XWSY15], auto-complete line drawings repetitions [XCW14], selection and grouping [XFAT12, NSS*12] and sketch beautification [LWH15, LRS18]. As the number of available 3D models and images steadily increases, effective methods for searching on databases have emerged. Using non-photorealistic rendering methods, meshes are transformed into sketches and search engines compute image descriptors that summarize global properties, such as contour histograms [PLR05], stroke similarity distance [SXY*11], Fourier transform [SI07], diffusion tensor fields [YSSK10] and bag-of-features models [ERB*12].

Learned descriptors gained popularity with the recent success of deep neural networks [LBH15]. Most applications involving line drawings, like Sketch Me That Shoe [YLS*16, SYS*17], target the problem of computing global descriptors for sketch-based image retrieval. Similarly, Qi et al. [QSZL16] and Bui et al. [BRPC17] proposed to train siamese networks that pulls feature vectors closer for sketch-image input pairs labelled as similar, and push them away if irrelevant. Zhu et al. [ZXF16] constructed pyramid cross-domain neural networks to map sketch and 3D shape low-level representations onto an unified feature space. Other authors investigated how to learn cross-modal representations that surpass sketch images and 3D shapes, incorporating text labels, descriptions, and even depth maps [TD16, CAV*16, ZRBL17]. Other learned descriptors applications include sketch classification and recognition [YYS*15, ZLZ*16]. Like Yu et al. [YYS*15], all these methods target global features that can discriminate high level characteristics in sketches a

single representation for an entire shape), our goal is to compute accurate pixel-wise descriptors that capture part semantics along with local and global contexts to perform local matching.

In the context of learning local semantic descriptors for photographs, a common strategy consists on training siamese architectures with pairs or triplets of corresponding and non-corresponding patches. Most of these methods [HLJ*15, ZK15, SSTF*15, CGSC16, KBCR16, TFW17] learn representations for natural image patches such that patches depicting the same underlying surface pattern tend to have similar representations. In contrast, we aim to learn a deep learning model able to assign similar descriptors to geometrically but also semantically similar points across different objects. Moreover, instead of a descriptor for a single image patch, our method learns a complex representation for a 3D surface point (depicted as a sketch) by exploiting the information from different views and multiple scales. Other proposals such as [KMH*17] learn a convolutional descriptor using self-similarity, called fully convolutional self-similarity (FCSS), and combine the learned descriptors with the proposal flow framework [HCSP16]. These approaches to learning semantic correspondences [ZKA*16] or semantic descriptors [HRH*17] generally perform better than traditional hand-crafted ones. However, since limited training data is available for semantic correspondence in photographs, these region-based methods rely on weakly-supervised feature learning schemes, leveraging correspondence consistency between object locations provided in existing image datasets. This makes them vulnerable to changes in orientation and projection distortion, and also to shape variation as commonly seen in line drawings, where the number and style of strokes may change significantly while the semantics of the parts are preserved.

Learned descriptors require adequate training datasets. The high diversity in style and the difficulty to automate sketch annotation makes it hard to compile massive line drawing datasets. Eitz et al. [EHA12] introduced a dataset of 20,000 sketches spanning 250 categories. Similarly, The Sketchy Database [SBHH16] ask crowd workers to sketch photographic objects sampled from 125 categories and acquired 75,471 sketches, compiling the first large-scale collection of sketch-photo pairs. Recently, Quick, Draw! [HE17] released an open source collection composed by 50 million doodles across 345 categories drawn by players of an online game. Nevertheless, the skills and style disparities of contributors to these datasets makes them unsuitable for our goal. Instead, we target design sketches that are drawn following approximately a particular set of rules [ES11]. Similar to Wang et al. [WKL15], we exploit shape collections augmented with semantic part-based correspondences data to synthesize sketches with NPR techniques. The registered 3D models naturally provide us with 2D/3D alignment, a crucial ingredient to learn our multidimensional features.

3. Multi-view Sketch Data

Shape collection. As with recent work targeting sketches and machine learning [DAI*17, HKYM16, SDY*18], we generated synthetic line drawings based on semantically corresponded 3D shapes. From the ShapeNetCore dataset [YKC*16] we selected models in 16 categories: airplane (1000), bag (152), cap (110), car (1000), chair (1000), earphone (138), guitar (1000), knife (784), lamp (1000), laptop (890), motorbike (404), mug (368), pistol (550), rocket (132), skateboard (304) and table (1000). The 3D models were augmented with correspondences files that provide a list of 10,000 randomly sampled surface matching points for every possible pair of shapes within each category. Correspondences are computed with a part-based registration algorithm that performs a non-rigid alignment of all pairs of segments with the same label over two target shapes as proposed in [HKC*18].

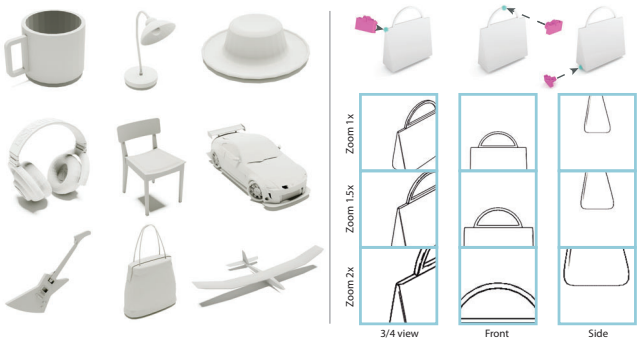

Figure 3: *Visualization of images in our dataset. Given a collection of almost 10,000 3D models distributed in 16 object categories, we rendered line drawings from three predefined angles and three different distances to the target surface point.*

Synthetic line drawings. While previous data-driven methods in sketching employ simple models such as Canny edges [1987] or image-space contours from [ST90], we adopted Apparent Ridges from [JDA07], a good approximation to artists lines as shown in *Where do people draw lines?* [CGL*12]. Apparent Ridges' lines approximate meaningful shape cues commonly drawn by humans to convey 3D objects. Apart from rendering style, viewport selection is crucial to convey shape in sketches. Literature in design recommends to adopt specific viewports in order to reduce the sketch ambiguity and simultaneously show most of the target shape [ES11]. Most 3D reconstruction algorithms from sketch images rely on assumptions like parallelism, orthogonality, and symmetry [CSE*16]. Following these guidelines, we selected a set of orthographic views to render from 3D models. We used a total of two accidental (object-aligned) views: front, right side and a single isometric angle, also called informative view: front-right (see Figure 3). Left sides were omitted since we assume that the objects are symmetric with respect to the front view (see Section 6). To capture images, we centred an orthographic camera on each sample point, and shoot it from three different constant distances and three distinct viewports (successive zooms at 1.0x, 1.5x and 2x). Occluded points were discarded by comparing z-buffer data with camera-to-target point distance. Rendering all the sampled surface points for each model would have been very computationally expensive. On the other hand, discarding some models from each category would not have been a good option, since the diversity of shapes favours generalization. For these reasons, we decided to randomly sub-sample surface points on each model, so that we only use a fraction of them. For every model in our dataset, we randomly choose and render 70 corresponding points to other models within its category. To increase diversity, we randomly select the sub-sampled points differently for each model.

In total, our dataset consists of 538,000 images in a resolution of 512×512 pixels. Each image has been augmented with the information needed to retrieve all other corresponding points in the dataset. It took approximately 15 days to complete the rendering stage on a PC equipped with an NVIDIA Titan Xp GPU and an Intel i7 processor. For the sake of reproducibility and to encourage further research, we aim to publicly release it by the time of publication.

4. Learning Multi-View Descriptors for Line Drawings

4.1. Proposed approach

Backbone architecture. Our proposed approach relies on a backbone convolutional neural network (CNN) that is responsible for learning and computing descriptors for each given input. We choose a CNN inspired by the standard AlexNet [KSH12], which comprises five convolutional layers, followed by ReLU non-linearities and max pooling (see supplemental material for details). Nevertheless, our method is sufficiently general to incorporate any other backbone architecture. In Section 5.4, we show SketchZooms performance when using other state of the art networks like VGG19 [SZ14] and ResNet-18 [HZRS16]. A key insight in our approach consists of aggregating local surface information across multiple zooms. Therefore, we modified all the aforementioned architectures to incorporate a pooling layer that aggregates the descriptors $Y_{z,p}, z \in Z$ generated for each of the three input zooms $X_{z,p}$ into a single one $Y_p = \max_z(Y_{z,p})$. The aggregation is performed on an element-wise maximum operation across the input zooms.

Loss function. A key component in our approach is the learning mechanism for tuning the network parameters. We adopted a triplet loss [SKP15] motivated by the fact that distances gain richer semantics when put into context, and the anchor point added by the triplet loss better shapes the embedding by exploiting this relativistic approach [WMSK17]. We strive for an embedding from a set of sketch image zooms $X_{z,p}$ centred on a point p, into a descriptor $Y_p \in \mathbb{R}^d$ ($d = 128$ in our setup). Triplet loss minimizes the distance between an anchor Y^a and a corresponding (also called positive) point descriptor Y^c. Simultaneously, it maximizes the distance between the anchor and a non-corresponding (negative) point descriptor Y^n. Formally, we want:

$$D^2(Y^a, Y^c) + \alpha < D^2(Y^a, Y^n), \tag{1}$$

where D stands for the Euclidean distance between descriptors, and α is a margin enforced between positive and negative pairs ($\alpha = 1$ in our implementation). Formulating Equation 1 as an optimization problem over the network parameters $\boldsymbol{w}$, we have:

$$\mathcal{L}(\boldsymbol{w}) = \sum_i^N \max\left(D^2\left(Y_i^a, Y_i^c\right) - D^2\left(Y_i^a, Y_i^n\right) + \alpha, 0\right), \tag{2}$$

where N is the cardinality of the triplets training set.

Naively using all triplets is highly inefficient since the more the training progresses, the more triplets are going to satisfy Equation 1, making training slower over time. Therefore, we adopted an alternative approach in which we adaptively select semi-hard triplets on each training step satisfying:

$$\left\{ \begin{array}{l} D^2\left(Y^a, Y^c\right) < D^2\left(Y^a, Y^n\right), \\ D^2\left(Y^a, Y^n\right) < D^2\left(Y^a, Y^c\right) + \alpha, \end{array} \right) \tag{3}$$

meaning that we look for training samples $\{Y^a, Y^c, Y^n\}$ lying inside the semi-hard margin area delimited by α. For the sake of notation, we refer to triplets using descriptor notation symbol Y. In practice, we compute $\{Y^a, Y^c, Y^n\}$ from input images using the last network training state. We build useful triplets on the fly for each training minibatch by testing whether their descriptors infringe Equation 3. In our setup, we cluster individual samples in groups G to be sequentially used during each training epoch. To build a minibatch, we randomly sample positive pairs from G of the form $[Y_i^a, Y_j^c],\ i, j \in G$. From construction, our dataset allows to easily obtain these corresponding pairs since they are exhaustively listed in custom files. We then test the semi-hard conditions over a random number s of negative samples $[Y_i^a, Y_k^n],\ i, k, \in G$. We experimented with several values for s and found $s = 5$ to minimize the time spent in random search while still providing good triplets for training.

4.2. Experimental setup

We experimentally evaluated multiple aspects of our approach: (i) we tested SketchZooms on a number of hand-drawn images from the OpenSketch dataset [GSH*19] to assess the generalization power of the network to unseen shape categories and styles, (ii) we examined the ability of our learned embeddings to properly distribute descriptors in the feature space, (iii) we computed correspondence accuracy metrics to evaluate matching performance in the image space, (iv) we performed a perceptual study to assess the semantic aspects of our features and (v) we compare the performance on the aforementioned metrics against other correspondences methods, with emphasis on those commonly applied to line drawings.

Metrics. We report quantitative results using two standard metrics. First, we tested our embedding space using cumulative match characteristic (CMC), a standard quality measure for image correspondences [KLR15, WN04]. This metric captures the proximity between points inside the embedding space by computing distances over descriptor pairs on two target sketches: given a point on one of the input images, a list of corresponding candidate matchings on the other image is retrieved; then, candidates are ranked using a proximity measure, e.g. the Euclidean distance in descriptor space. We also evaluated the accuracy of our descriptors on the image space using the correspondence accuracy (CAcc) from [KLM*13], over our set of test samples. This metric evaluates the accuracy of predicted correspondences with respect to the ground truth by registering all L2 distances between retrieved matching points and ground truths. We report the percentage of matchings below normalized euclidean distance (5% of image side (512 pixels)).

Competing descriptors. We compare our method against state-of-the-art descriptors commonly used for local sketch matching tasks, including the radial histograms from ShapeContext [BMP02] and the GALIF descriptor, based on Gabor filters by Eitz et al. [ERB*12]. We additionally consider a hand-crafted descriptor consisting on principal component analysis (PCA) over a small neighbourhood of pixels surrounding the target point. We also

compared SketchZooms against deep learned features. In particular, we considered MatchNet [HLJ*15], a patch-based descriptor targeting natural images, and the multiview architecture from Huang. et al. [HKC*18] to compute local 3D shape descriptors. The latter is closely related to our work, although it relies on a contrastive instead of a triplet loss, and does not apply our hard samples mining strategy during training. To the best of our knowledge, no deep learning based approaches have been introduced specifically for local sketch matching tasks. For all the aforementioned methods, we used the official and freely available implementations when possible, or re-implemented them otherwise. Importantly, for a fair comparison, all deep learning-based methods backbones were adapted to work with AlexNet and retrained with our synthetic sketch dataset.

Data augmentation and training details. We computed local descriptors from a set of zoomed sketch views Z (three in our setup) centred on the point of interest p. The network learns rotational invariant descriptors by randomly rotating input images between 0 and 360 degrees with equal probability. To keep the descriptor robust to different resolutions, we downsampled the input image size by 30% and 60% with a probability of 0.2. To diminish sensitivity to the camera-target point distance, we added noise during training to the zoom parameter by sampling camera displacements from a normal distribution (with $\mu = 0$ and $\sigma^2 = 0.3$, where 0.3 means 30% size increment w.r.t. the original image size). Since some views are more densely populated than others, we restrict our training minibatches to have the same number of samples from each view in order to avoid bias. Also, since each object class has a different total number of images, we restricted each batch to equally balance the amount of images from each category. Our data augmentation choices were iterative, and empirically guided by results obtained during the experimentation stage using a validation set.

The network architecture was implemented with PyTorch and trained on NVIDIA Titan Xp GPUs. We first initialize the convolutional layers using AlexNet weights trained on the ImageNet dataset, as provided in Pytorch. The learning rate was set to $l = 10^{-5}$ and the network was trained for 185 epochs. We optimize the objective in Equation 2 using Adam optimization [KB14] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a batch size of 64 triplets. We did not use batch normalization layers or dropout in addition to those already into AlexNet (dropout $p = 0.5$ on layer fc6).

5. Results

5.1. OpenSketch benchmark

In order to evaluate our learned features, we conducted a series of comprehensive comparisons against other methods when applied to hand-drawn design sketches. We relied on OpenSketch [GSH*19], a dataset of product design sketches containing more than 400 images representing 12 man-made objects drawn by 7 to 15 product designers of varying expertise. These design sketches are drawn in a variety of styles and from very different viewports. In addition, all images are augmented with a series of corresponding points derived from registered 3D models and manually annotated by designers. Importantly, none of the 12 OpenSketch object categories match those in our training dataset: bumps, hairdryer, mixer, potato chip, tubes, waffle iron, flange, house, mouse, shampoo bottle, vacuum cleaner and wobble surface. This is a key factor to assess the descriptors' generalization power, particularly those which are learned from our synthetic training data. Since each line drawing in the dataset has several layers at different progress stages, we filtered them and kept the latest version of the sketch (called *presentation* sketch). Additionally, since sketches drawn from observation tend to be aligned with the horizontal axis, we altered them by applying a random rotation of $\pm 90^\circ$ to each image before computing the descriptors. In this way, we effectively evaluate each method's ability to build rotation invariant descriptors. The correspondences between all image pairs were computed using the Euclidean distance in descriptor space, and choosing the closest target point on each case.

Figure 4 shows the retrieved matchings on image pairs from different artists in the dataset for all evaluated methods. Overall, our approach was able to successfully exploit the features learned from the synthetic training set when working with hand-drawn images. We quantitatively evaluated descriptors on this benchmark by computing the correspondences among all possible pairs of images within each category, a total of 66,320 corresponding points. Table 1 reports the performance of the evaluated descriptors over five retrieved matches for the CMC and below 5% normalized Euclidean distance for the CAcc. We further illustrate these metrics in Figure 5. Our descriptors outperformed the competing methods in both evaluated metrics and across all object categories. According to the reported metrics, we observed that our learned descriptors outperform the rest, including the patch-based learned descriptors of MatchNet[HLJ*15] and the multi-view architecture of Huang et al. [HKC*18]. Also, SketchZooms performed better than the hand-engineered local descriptors traditionally used for corresponding line drawings, namely ShapeContext [BMP02] and GALIF [EHA12]. Following these results, we believe that our method can successfully embed semantically similar feature points in descriptor space closer than other methods, while being stable to changes in view, decorations, and style. Moreover, despite the fact that testing categories differ from those used for training, our method can still exploit 3D shape cues to produce fairly general local descriptors that perform favourably compared to general hand-crafted alternatives.

5.2. Perceptual study

Humans possess an extraordinary ability to resolve semantic correspondences in multi-view scenarios thanks to their previously-acquired 3D knowledge about the world. We conducted a perceptual study to comprehensively assess the relationship between the semantics captured by our descriptors from synthetic data and the decisions made by humans when performing the same matching task on artificial sketches. Each of the 10 study volunteers was presented with $m = 4$ points on a synthetic sketch image (origin), and was instructed to find m corresponding points on a target image. We used a total of 40 random image pairs from our synthetic dataset distributed in four categories: bag, chair, earphone and mug. Points on the origin images were randomly selected from a larger list of feature points computed over all the study images using the corner detector Good Features to Track [ST93]. For the target points, we used blue noise sampling to distribute 200 candidate points across the image. We did not show any of these candidates to the study participants. Instead,

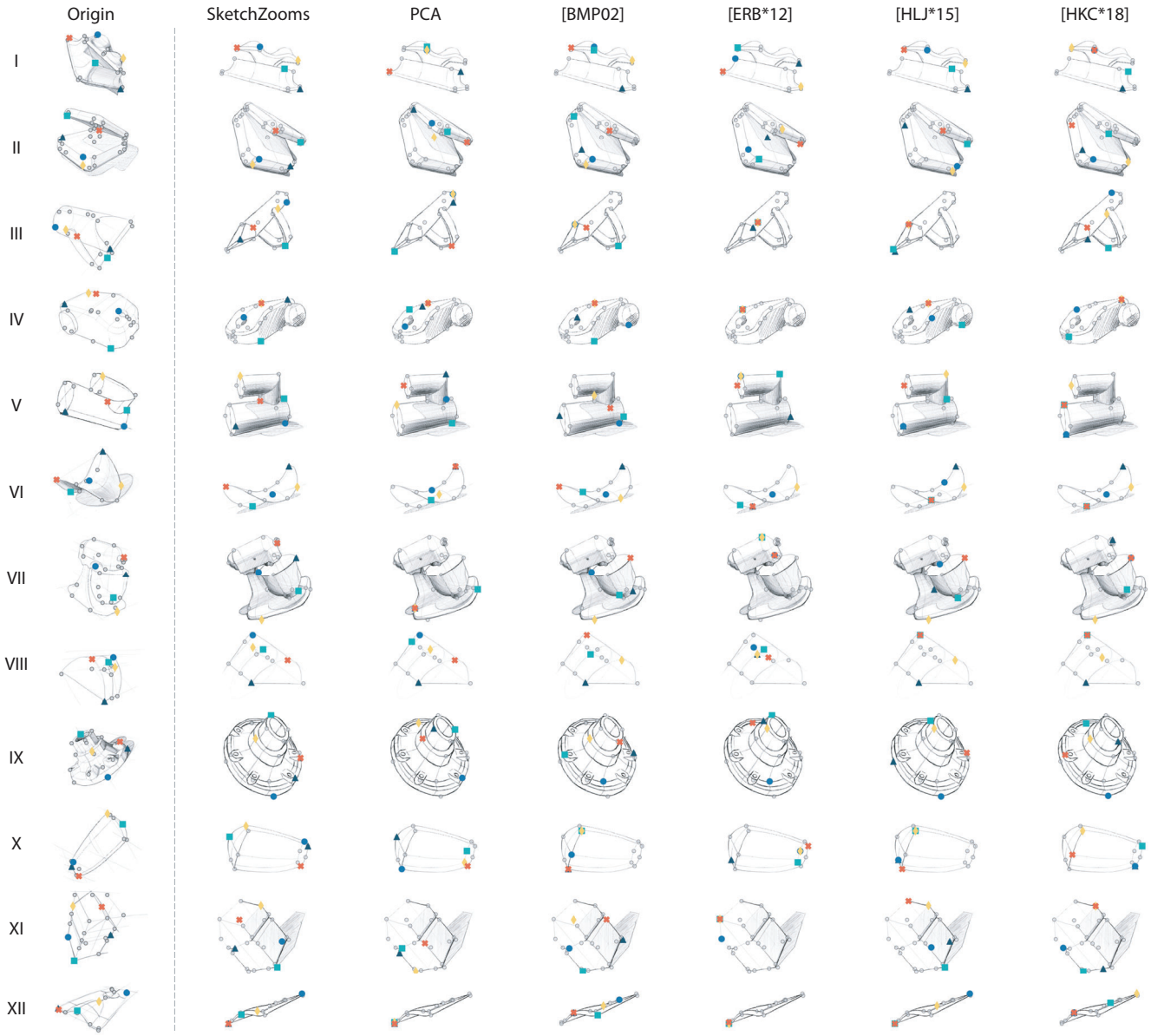

Figure 4: *Pair-wise sparse correspondences on OpenSketch data. Images have been randomly rotated between* $\pm 90^{\circ}$ *before computing the descriptors. Columns show origin and target images, corresponded using different local descriptors. For each image pair, we highlighted five points distributed in different areas of the image. The grey dots show the remaining sampled points for which matches were obtained in order to compute metrics in Table 1. Only Euclidean distance in feature space has been considered to determine these correspondences. Overall, our learned descriptors manage to identify similar underlying local shapes, despite the extreme differences in style, hatching, shadows, construction lines and camera positions.*

and similar to the approach adopted by BestBuddies [ALS*18], we registered mouse clicks over targets, and fitted 2D Gaussian distributions over the coordinates annotated by the users. Overall, we observe all participants consistently corresponded points on target images within specific regions. After the subjects sessions, we retrieved matchings among origin and target images by selecting the closest points in the Euclidean space for all compared methods. We then defined a similarity measure by evaluating the average fitted probability density function on the top five retrieved matches for each query point. Higher similarity scores are then assigned to regions where the consensus among users and the automatically retrieved points is strong, and vice-versa. We averaged the scores for all the points within each object category and summarized the results in Table 2.

Table 1: *Cumulative match characteristic (percentage of correct matches obtained in the top five rank), and correspondence accuracy (percentage of matchings below 5% of the image width in Euclidean space w.r.t. ground truth) on all evaluated methods over OpenSketch dataset. In total, our test samples consist of 66,320 corresponding points.*

	mixer		tubes		wobble surface		hairdryer		vacuum cleaner		mouse	
	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc
[BMP02]	42.51%	20.24%	44.79%	19.56%	38.15%	20.64%	49.62%	22.58%	31.11%	10.98%	62.74%	21.35%
[ERB*12]	24.55%	11.99%	30.87%	9.97%	35.84%	17.36%	32.85%	13.30%	27.68%	10.47%	55.94%	13.27%
[HLJ*15]	41.78%	18.51%	48.41%	19.90%	36.89%	18.29%	45.33%	19.66%	34.85%	12.78%	60.72%	19.48%
[HKC*18]	40.20%	16.72%	49.39%	16.29%	33.63%	16.32%	42.87%	18.30%	35.95%	11.85%	60.55%	17.87%
PCA	20.46%	9.73%	30.63%	9.52%	30.95%	16.92%	26.84%	10.46%	26.43%	9.88%	49.65%	11.13%
SketchZooms	**62.67%**	**37.28%**	**66.44%**	**31.90%**	**50.70%**	**31.49%**	**57.71%**	**31.65%**	**49.07%**	**24.39%**	**71.36%**	**27.86%**

	bumps		potato chip		shampoo botle		waffle iron		flange		house	
	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc
[BMP02]	40.16%	21.02%	58.75%	26.49%	65.26%	28.90%	19.16%	9.47%	46.87%	18.12%	48.04%	17.55%
[ERB*12]	32.07%	12.28%	45.32%	14.16%	44.43%	16.31%	15.68%	7.28%	29.55%	10.79%	39.30%	11.84%
[HLJ*15]	39.65%	18.39%	60.72%	24.20%	63.25%	23.80%	19.06%	10.99%	40.21%	16.54%	44.33%	14.79%
[HKC*18]	36.63%	15.90%	55.16%	19.27%	62.03%	23.73%	18.62%	10.34%	38.81%	13.29%	40.88%	11.55%
PCA	30.94%	14.35%	43.98%	12.58%	38.78%	15.76%	14.14%	7.67%	23.23%	7.73%	26.55%	6.57%
SketchZooms	**45.98%**	**25.40%**	**66.63%**	**30.27%**	**70.08%**	**34.73%**	**23.66%**	**13.59%**	**52.15%**	**24.93%**	**51.80%**	**21.93%**

Table 2: *Perceptual study metrics.*

	bag	chair	earphone	mug
[BMP02]	0.023±0.013	0.024±0.008	0.021±0.011	0.022±0.010
[ERB*12]	0.005±0.003	0.006±0.004	0.009±0.006	0.008±0.004
[HLJ*15]	0.014±0.010	0.019±0.005	0.016±0.006	0.018±0.010
[HKC*18]	0.011±0.005	0.015±0.008	0.014±0.003	0.013±0.007
PCA	0.003±0.002	0.006±0.004	0.006±0.003	0.004±0.004
SketchZooms	**0.025±0.007**	**0.025±0.009**	**0.024±0.009**	**0.024±0.012**

Figure 6 illustrates matchings computed with each local descriptor and the areas where the subjects consensus was stronger. When the participants had to disambiguate between points with identical semantics in symmetric views of an object (Figure 6 II, III and IX), most of them decided to choose those on the same relative position with respect to its counterpart in the origin image. SketchZooms descriptors often find multiple semantically similar candidates on both sides of the vertical symmetry plane (such in rows II, III and IV from Figure 6). This aspect of our descriptors make them more robust to arbitrary rotations and reflections, as shown in Section 5.1. An extended discussion about symmetry is presented in Section 6.

Correspondences obtained with our descriptors are closer to hot areas than those produced with other methods. ShapeContext also produced accurate descriptors for these images. However, this is contradictory with the performance previously observed on the OpenSketch benchmark. We believe this difference is likely due to ShapeContext lacking a learning strategy, which renders a method unable to generalize to more complex, realistic sketch data such as OpenSketch. ShapeContext is a hand-crafted descriptor designed to correspond shape outlines that look similar and clean, like the synthetic sketches used in the study. When corresponding hand-drawn images with severe projection distortions, multiple rough strokes and shading, ShapeContext fails to recognize the underlying similar local shapes (Figure 4). The full set of images from subjects data is available as supplementary material.

5.3. Triplet vs. contrastive loss

Similar to our approach, the work by Huang et al. [HKC*18] relies on a multi-view architecture to learn descriptors for 3D models, trained using a contrastive loss, and random minibatches built during the learning phase without using any sampling heuristic. On the other hand, our work relies on a triplet loss function and uses a custom training schedule that can potentially benefit other application dealing with unbalanced sets of views. In order to empirically compare these two approaches, we trained an adaptation of the approach of Huang et al. to this specific problem, keeping all training hyperparameters and the aforementioned data augmentation strategies to avoid mixing effects in the evaluations. SketchZooms performed better on all testing categories (Table 1), with average CMC = 55.52% (SketchZooms) over CMC = 42.89% ([HKC*18]), and CAcc = 27.95% (SketchZooms) over CAcc = 17.68% ([HKC*18]). These experiments support our hypothesis that a combined training strategy based on a triplet loss and a smart data sampling procedure is of paramount importance in order to improve results with respect to basic contrastive losses and random samplings. Additionally, results indicate that multi-view convolutional neural network architectures can learn meaningful semantic descriptors in contexts where the texture image information is very scarce and ambiguous, like line drawings.

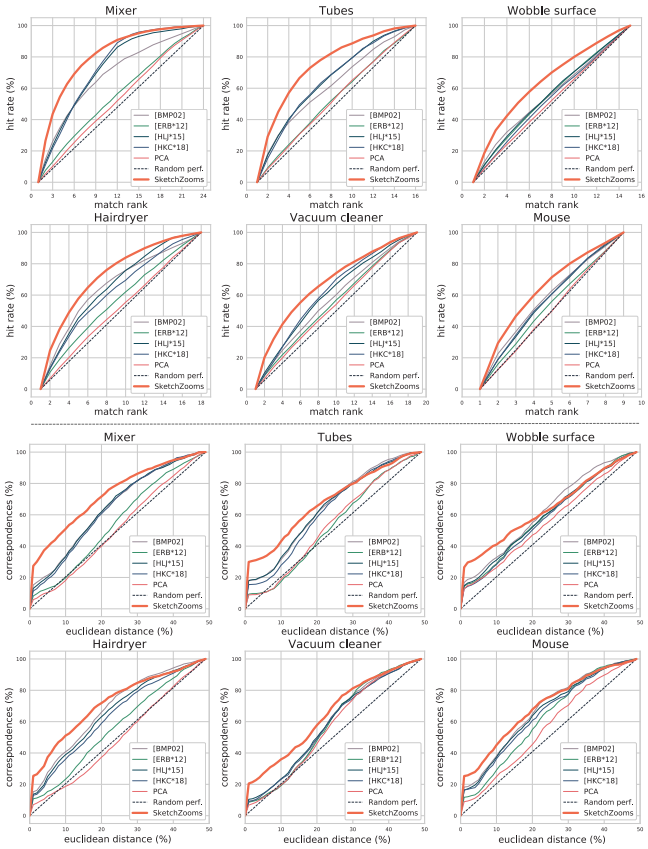

Figure 5: *Top: Cumulative match characteristic plots for the evaluated descriptors on the test dataset for mixer, tubes, wobble surface, hairdryer, vacuum cleaner and mouse categories. y-axis accounts for the percentage of matchings retrieved below the raking position indicated on x. Bottom: Correspondence accuracy curves, where x-axis shows normalized Euclidean distance error, and y-axis accounts for the matching percentage of retrieval below the error margin indicated on x.*

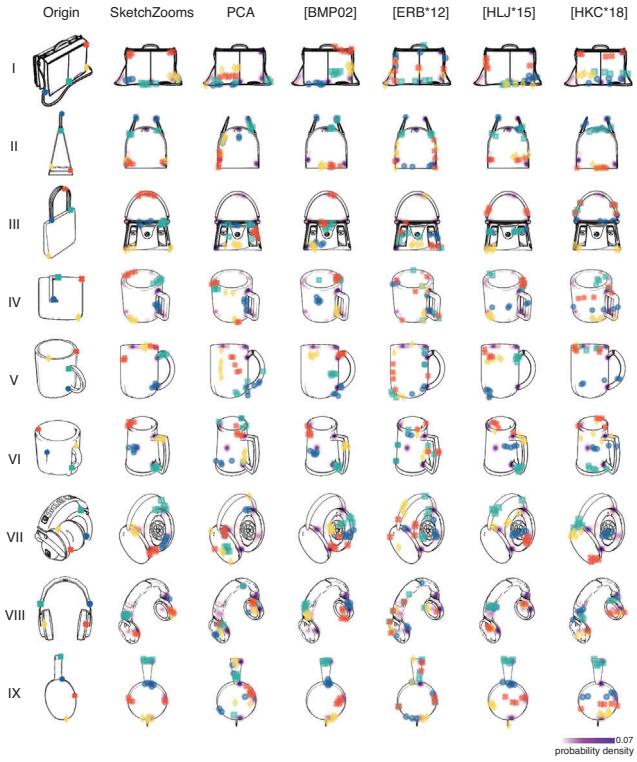

Figure 6: *Perceptual study. On each row, the first column shows the origin image together with the four sampled points shown to participants of the study. All other columns show the top five retrieved correspondences computed with each local descriptor among a total of 200 target points. The heatmaps underneath show the probability density distribution of the subjects clicks. The remaining image data from the study is available as supplementary material.*

5.4. Architecture alternatives

We further investigated the effect of adopting other network architecture as backbones in our pipeline. Therefore, we trained our framework using two alternative architectures, namely VGG19 [SZ14] (133,387,752 total parameters) and ResNet-18 [HZRS16] (11,242,176 total parameters). Both models were fine-tuned from ImageNet weights using the same hyperparameter settings as AlexNet (40,796,610 total parameters), with the only exception of the batch size for VGG19, which had to be reduced by half due to the large memory requirements of the network. Table 3 summarizes the performance over the OpenSketch benchmark data. Overall, we found a slight improvement on the evaluated metrics across most object categories when using VGG19 architecture. We believe this is likely due to the well-known properties of VGG19 as a feature extractor, observed in multiple different applications [SRASC14]. It must be noticed, however, that these advantages come at the cost of a much slower training due to the significant amount of parameters on this network, most of them originated in the last series of fully connected layers. ResNet-18, on the other hand, performed much more poorly in our experiments, probably due to the lack of a stack of fully connected layers and the usage of global average pooling.

6. Robustness and Limitations

We now discuss the overall behavior of our method under challenging scenarios and its main limitations.

Robustness to sketchiness. Adopting *Apparent Ridges* as our dataset rendering engine allowed our method to be robust to typical drawings' sketchiness. Synthetic images rendered with this method often contain wiggly lines and several other imperfections. Our experimental setup allowed us to evaluate SketchZooms ability to deal with very different drawing styles (Figure 4, V and VII), even with overlayed construction lines and shadows (Figure 4, IX and XI). However, extremely rough drawings, with an excessive amount of construction lines or extreme lighting can harm the performance of our descriptors (i.e. cross-hatched areas in houses from Figure 7).

Symmetry. Most man-made objects are symmetric with respect to at least one plane in 3D space. Our features are strongly biased by the image semantic information, and sometimes can mismatch

Table 3: *Ablation study metrics for the OpenSketch benchmark. Cumulative match characteristic (top five rank) and correspondence accuracy for normalized Euclidean distance at 5% on all evaluated backbone architectures.*

	mixer		tubes		wobble surface		hairdryer		vacuum cleaner		mouse	
	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc
ResNet-18	60.98%	33.83%	63.99%	29.89%	47.89%	29.97%	51.43%	27.27%	45.25%	20.85%	68.82%	23.82%
VGG-19	**63.95%**	37.15%	**69.66%**	**36.16%**	49.60%	30.92%	**57.91%**	31.59%	48.62%	23.11%	**72.68%**	27.58%
AlexNet	62.67%	37.28%	66.44%	31.90%	**50.70%**	**31.49%**	57.71%	**31.65%**	**49.07%**	**24.39%**	71.36%	**27.86%**

	bumps		potato chip		shampoo botle		waffle iron		flange		house	
	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc	CMC	CAcc
ResNet-18	43.16%	21.54%	65.64%	28.20%	71.36%	32.95%	23.40%	13.52%	51.44%	21.54%	51.32%	19.32%
VGG-19	44.97%	**25.43%**	**71.08%**	**32.92%**	**72.10%**	34.17%	**24.49%**	**14.65%**	**52.73%**	23.57%	**55.76%**	**24.26%**
AlexNet	**45.98%**	25.40%	66.63%	30.27%	70.08%	**34.73%**	23.66%	13.59%	52.15%	**24.93%**	51.80%	21.93%

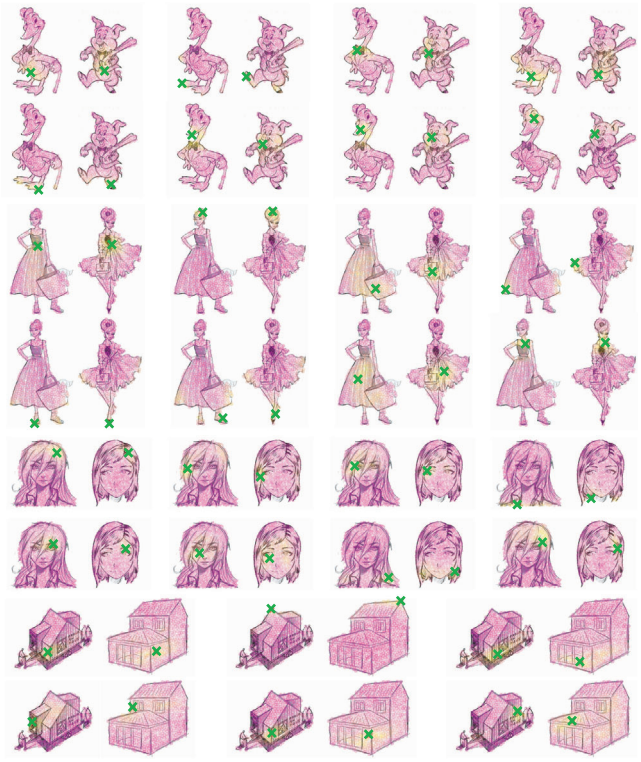

Figure 7: *SketchZooms dense correspondences on line drawings from different styles (from top to bottom: cartoons, fashion, manga and architectural sketches). Colours indicate distance to target point (green cross) in feature space. For each line drawing, between 400 and 1000 points were randomly sampled and corresponded using Euclidean distance.*

on symmetric points on the target sketch (see Figure 6). Symmetry mismatches happen more often when trying to correspond extreme viewports, like the side and front of two target objects. For all results reported in this paper, we used the Euclidean distance in feature space to retrieve correspondences and compute metrics. However, simultaneously matching several points can help disambiguate these symmetries, -i.e. combinatorial optimization methods like the Hungarian algorithm [Kuh55] can help refine more coherent matchings than using a simple strategy of matching closer points in descriptor space. An interesting future research direction is to explore ways to incorporate orientation tags in the training phase or to involve users actively in refining correspondences on the fly.

Zoom and rotation sensitivity. As mentioned in Section 4, in order to compute a descriptor for a given image point, we need to successively crop three zoomed images surrounding it. These images are aggregated and transformed by the SketchZooms network to produce a descriptor of the point. We pick the zoom parameter value in order to include some information of the strokes composing the target image, since providing three empty images to the network would produce undesired outputs. In particular, for all results presented in this paper we fixed zoomed images sides to be 10%, 20% and 40% of the total image length (512 pixels in our experiments). In general, OpenSketch images are relatively on the same scale, occupying at least two thirds of the total width and aligned with the horizontal image plane. To assess the effect of different zooms and rotations, we performed a controlled study where the testing images were zoomed in or out at different scales before computing the descriptors. In particular, we segmented the objects from OpenSketch images and re-scaled them randomly at different maximum sizes $\pm 10\%$, $\pm 20\%$ and $\pm 40\%$. We measured size as the maximum distance among all pairs of stroke pixels for each image. We also generated versions of the dataset where images were randomly rotated up to $\pm 45^\circ$, $\pm 90^\circ$, and $\pm 180^\circ$. Then, we computed the evaluated metrics on all possible corresponding pairs within each category. Table 4 summarizes the results. While SketchZooms performance is not greatly affected by these parameters, zooming too much can lead to cases in which the three cropped images have any stroke information, while zooming too little could miss details, degrading the output descriptor quality.

Generalization to unseen line drawing styles. Finally, we show the capability of SketchZooms to perform on images significantly different than the ones used for training. We selected pairs of sketches from the Yan et al. [YVG20] public dataset and computed dense correspondences. Figure 7 shows exemplary outputs of corresponding points in cartoons, manga, fashion and architectural

Table 4: *Descriptors performance under random zoom and rotations up to the values indicated in top rows for each table. Cumulative match characteristic is reported for the top five rank and correspondence accuracy for normalized Euclidean distance at 5% of the image width.*

	±10%		±20%		±40%	
max. zoom	CMC	CAcc	CMC	CAcc	CMC	CAcc
[BMP02]	46.69%	20.44%	44.85%	19.38%	44.73%	20.38%
[ERB*12]	34.31%	12.61%	33.75%	12.31%	34.02%	13.17%
[HLJ*15]	45.42%	18.67%	44.16%	17.56%	43.18%	18.07%
[HKC*18]	43.04%	16.82%	41.68%	15.94%	42.43%	16.95%
PCA	30.09%	11.07%	30.43%	10.91%	30.21%	11.93%
SketchZooms	**55.39%**	**27.52**	**53.97%**	**26.69%**	**52.93%**	**26.78%**
	±45°		±90°		±180°	
max. rotation	CMC	CAcc	CMC	CAcc	CMC	CAcc
[BMP02]	51.53%	24.79%	45.60%	19.74%	38.06%	15.10%
[ERB*12]	36.79%	14.27%	34.51%	12.42%	31.58%	10.97%
[HLJ*15]	49.37%	21.22%	44.51%	18.11%	35.83%	16.66%
[HKC*18]	54.78%	17.39%	42.89%	15.95%	34.91%	12.52%
PCA	30.23%	11.32%	30.21%	11.03%	30.19%	10.90%
SketchZooms	**59.38%**	**31.14%**	**55.69%**	**27.95%**	**44.58%**	**20.72%**

sketches. Overall, our learned features produced plausible matchings. Importantly, the distance field in feature space reveals a smooth embedding, where semantically and geometrically similar points are close to each other. This smoothness does not appear to be significantly altered by the rough shading variation and other discontinuities in the images. Even if none of these sketch categories were used to train our model, our highly diverse synthetic dataset used for training ensured a regularization effect, allowing generalization to unseen styles.

7. Applications

Image morphing for shape exploration. Inspired by the recent work of Arora et al. [ADN*17], we implemented an image morphing algorithm based on the image mapping obtained from the SketchZooms features. The goal is to allow exploration of the continuous design space between two sketches while smoothing views and shape transitions. We start by computing motion paths between sparse SketchZooms corresponding points, and then interpolate them into dense smooth trajectories. We sample $k = 10$ correspondences evenly distributed over the input-target pair. Then, we compute a Delaunay triangulation of the image space using the sampled points as input. For each triangle, we estimate an affine transformation that maps both triangulations on a number of steps $\mathbf{s} = 50$. We implemented a non-linear alpha blending function to reduce ghosting effects for a pixel p at a step $\mathbf{s}$ defined as:

$$\alpha_p(\mathbf{s}) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{\mathbf{s} - \delta(p)}{\rho(p)}\right), \tag{4}$$

where δ and ρ are linear functions of the pixel confidence score to keep the sigmoid outputs in the [0,1] interval. This blending function ensures that well matched regions smoothly transition into other images, while regions with poor matching disappear quickly from the image (Figure 8).

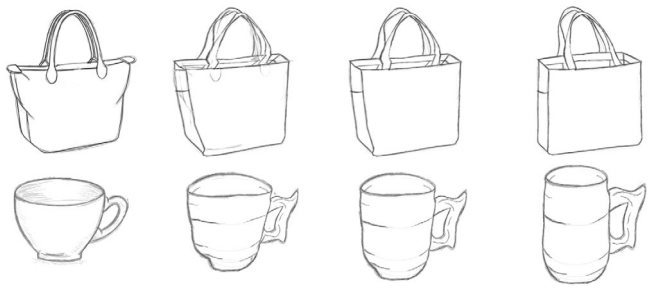

Figure 8: *Image morphing sequences using SketchZooms descriptors for corresponding two target sketches. A non-linear alpha blending map was computed from point distances in the SketchZooms feature space.*

Part segmentation. Sketch segmentation has been addressed before as an instance of colourization [SDC09] and simplification [NSS*12, LRS18]. Segmentation can be used for different applications, like adding depth information to line drawings or applying global illumination effects [SSJ*10, SKČ*14]. Similarly, SketchZooms' features can be used to perform automatic semantic layering and colouring, since painting has much in common with image segmentation. Specifically, we first manually segmented hand-drawn images from the headphone category (10 in our test application). Then, we computed SketchZooms' features for a subset of 2D points on every sketch using blue noise sampling, and used them to train a *C-SVM* classifier which learns to predict labels from our descriptors. Formally, we solve for:

$$\begin{gathered} \min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \\ \text{subject to } y_i\left(w^T \phi(x_i) + b\right) \geq 1 - \zeta_i \\ \zeta_i \geq 0, i = 1, \ldots, n \end{gathered}, \tag{5}$$

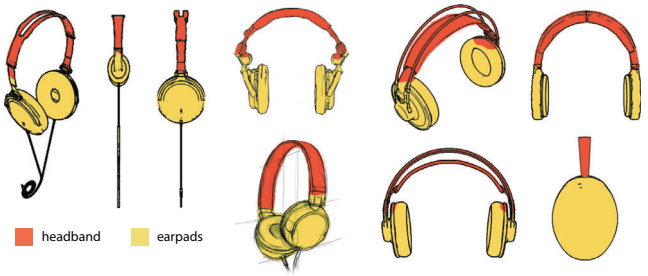

Figure 9: *SketchZooms features re-purposed for semantic segmentation. Labels can be used to decompose sketches into different layers or as pixel-wise semantic tags for colouring.*

Figure 10: *Results from our 3D shape search engine. Even though the searched models had no ground truth correspondence on the model database, our algorithm returned plausible shapes. Our features additionally provide information about the sketch view, allowing to automatically orient models to the query sketch.*

where C is the capacity constant (set to $C = 1$), $\boldsymbol{w}$ is the vector of coefficients, and $\boldsymbol{\zeta}_i$ represents parameters for handling non-separable data. The index i labels the n training cases ($n = 2$ in our setup). Figure 9 shows the semantic segmentations obtained with our classifier.

Sketch-based 3D shape retrieval. As shown in Section 2, much of the work on image features for sketches was proposed in the context of 3D retrieval applications. In order to test the potential of our features in this task, we implemented a 3D model search engine based on our local descriptors. We computed SketchZooms descriptors for 70 random point samples over the synthetic line drawings of 70 earphone models (4900 points in total distributed among 3 different viewports). At searching time, we sample 1000 points from query sketches using blue noise sampling, and retrieve the candidate model list using L2 distance w.r.t. query points. This simple strategy retrieves similar models in the database (Figure 10). Additionally, our search engine can accurately determine which camera viewport best matches the query sketch in order to consistently orient 3D models, demonstrating the capability of our feature vectors to encode viewport information.

8. Conclusions

We presented SketchZooms, a learnable image descriptor for corresponding sketches. To the best of our knowledge, SketchZooms is the first data-driven approach that automatically learns semantically coherent descriptors to match sketches in a multi-view context. Aiming this with deep neural networks was unfeasible before due to data limitation, as massively collecting sketches from artists and designers is extremely challenging. We have put together a vast collection of synthetic line drawings from four human-made objects categories and camera viewports commonly adopted by designers. This dataset can be easily extended with our pipeline as more 3D models become available. More importantly, our learned features were able to generalize to sketches in the wild directly from the synthetic data.

Our results offer interesting future directions of research. Apart from the already mentioned applications, like 3D part segmentation, semantic morphing and sketch-based retrieval, more technical research venues are also raised by this proposal. It is relevant to investigate whether other viewport configurations are possible without introducing much ambiguity into the descriptor space. Also, recent approaches have proposed to use semi-supervised hand-drawn images to improve network performance [SSII18]. Investigating whether explicit treatment of domain shifts can boost performance on our hand-drawn data set is an interesting future direction to explore. Finally, a deep study on how humans perform matching tasks on the sketch image domain would be very beneficial to build more accurate descriptors.

Acknowledgements

This study was supported by Agencia Nacional de Promoción Científica y Tecnológica, Argentina, PICT 2018-04517, PID UTN 2018 (SIUTNBA0005139), PID UTN 2019 (SIUTNBA0005534) and NVIDIA GPU hardware grant that provided two Titan Xp graphic cards. We also want to deeply acknowledge the short research internship program from Universidad Nacional de la Patagonia San Juan Bosco, Argentina, and all the perceptual study volunteers.

References

Arora R., Darolia I., Namboodiri V. P., Singh K., Bousseau A.: Sketchsoup: Exploratory ideation using design sketches. *Computer Graphics Forum 36* (2017), 302–312.

Aberman K., Liao J., Shi M., Lischinski D., Chen B., Cohen-Or D.: Neural best-buddies: Sparse cross-domain correspondence. *arXiv preprint arXiv:1805.04140* (2018).

Belongie S., Malik J., Puzicha J.: *Shape Matching and Object Recognition Using Shape Contexts*. Tech. rep., Deptartment of Computer Science and Engineering, California University, San Diego, La Jolla, 2002.

Bui T., Ribeiro L., Ponti M., Collomosse J.: Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *Computer Vision and Image Understanding 164* (2017), 27–37.

Castrejon L., Aytar Y., Vondrick C., Pirsiavash H., Torralba A.: Learning aligned cross-modal representations from weakly aligned data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2940–2949.

Chen T., Cheng M.-M., Tan P., Shamir A., Hu S.-M.: Sketch2photo: Internet image montage. *ACM Transactions on Graphics (TOG) 28*, (2009), 124.

Cole F., Golovinskiy A., Limpaecher A., Barros H. S., Finkelstein A., Funkhouser T., Rusinkiewicz S.: Where do people draw lines? *Communications of the ACM 55*, 1 (2012), 107–115.

Choy C. B., Gwak J., Savarese S., Chandraker M.: Universal correspondence network. In *Advances in Neural Information Processing Systems* (2016), pp. 2414–2422.

Cordier F., Singh K., Etem E., Cani M.-P., Gingold Y.: Sketch-based modeling. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Tutorials* (2016), Eurographics Association, p. 7.

Delanoy J., Aubry M., Isola P., Efros A. A., Bousseau A.: 3d sketching using multi-view deep volumetric prediction. *arXiv preprint arXiv:1707.08390* (2017).

Eitz M., Hays J., Alexa M.: How do humans sketch objects? *ACM Transactions on Graphics 31*, 4 (2012), 44–1.

Eitz M., Richter R., Boubekeur T., Hildebrand K., Alexa M.: Sketch-based shape retrieval. *ACM Transactions on Graphics 31*, 4 (2012), 31–1.

Eissen K., Steur R.: Sketching: The Basics (ed. 2012) amsterdam. *Google Scholar* (2011).

Gryaditskaya Y., Sypesteyn M., Hoftijzer J. W., Pont S., Durand F., Bousseau A.: Opensketch: A richly-annotated dataset of product design sketches. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 232.

Ham B., Cho M., Schmid C., Ponce J.: Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 3475–3484.

Ha D., Eck D.: A neural representation of sketch drawings. *arXivpreprint arXiv:1704.03477* (2017).

Hertzmann A.: Why do line drawings work? A realism hypothesis. *Perception* (2020), 0301006620908207.

Huang H., Kalogerakis E., Chaudhuri S., Ceylan D., Kim V. G., Yumer E.: Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics (TOG) 37*, 1 (2018), 6.

Huang H., Kalogerakis E., Yumer E., Mech R.: Shape synthesis from sketches via procedural models and convolutional networks. *IEEE Transactions on Visualization and Computer Graphics 2* (2016).

Han X., Leung T., Jia Y., Sukthankar R., Berg A. C.: Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3279–3286.

Han K., Rezende R. S., Ham B., Wong K.-Y. K., Cho M., Schmid C., Ponce J.: Scnet: Learning semantic correspondence. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 1831–1840.

He K., Zhang X., Ren S., Sun J.: Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.

Iarussi E., Bousseau A., Tsandilas T.: The drawing assistant: Automated drawing guidance and feedback from photographs. In *ACM Symposium on User Interface Software and Technology (UIST)* (2013), ACM.

Judd T., Durand F., Adelson E.: Apparent ridges for line drawing. In *ACM Transactions on Graphics (TOG) 26*, (2007), 19.

Kingma D. P., Ba J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Kumar BG V., Carneiro G., Reid I.: Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 5385–5394.

Kim V. G., Li W., Mitra N. J., Chaudhuri S., DiVerdi S., Funkhouser T.: Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 70.

Karanam S., Li Y., Radke R. J.: Person re-identification with discriminatively trained viewpoint invariant dictionaries. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 4516–4524.

Kim S., Min D., Ham B., Jeon S., Lin S., Sohn K.: FCSS: fully convolutional self-similarity for dense semantic correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 6560–6569.

Krizhevsky A., Sutskever I., Hinton G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.

Kuhn H. W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, *1-2* (1955), 83–97.

LeCun Y., Bengio Y., Hinton G.: Deep learning. *Nature 521*, 7553 (2015), 436.

Lamb A., Ozair S., Verma V., Ha D.: Sketchtransfer:A challenging new task for exploring detail-invariance and the abstractions learned by deep networks. *arXiv preprint arXiv:1912.11570* (2019).

Lowe D. G.: Object recognition from local scale-invariant features. In *Computer vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (1999), vol. 2, IEEE, pp. 1150–1157.

Liu C., Rosales E., Sheffer A.: Strokeaggregator: Consolidating raw sketches into artist-intended curve drawings. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 97.

Liu X., Wong T.-T., Heng P.-A.: Closure-aware sketch simplification. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 168.

Noris G., Sỳkora D., Shamir A., Coros S., Whited B., Simmons M., Hornung A., Gross M., Sumner R.: Smart scribbles for sketch segmentation. *Computer Graphics Forum 31* (2012), 2516–2527.

Pu J., Lou K., Ramani K.: A 2d sketch-based user interface for 3d cad model retrieval. *Computer-aided Design and Applications 2*, 6 (2005), 717–725.

Qi Y., Song Y.-Z., Zhang H., Liu J.: Sketch-based image retrieval via siamese convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on* (2016), IEEE, pp. 2460–2464.

Sangkloy P., Burnell N., Ham C., Hays J.: The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 119.

Sỳkora D., Dingliana J., Collins S.: Lazybrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum 28* (2009), 599–608.

Su W., Du D., Yang X., Zhou S., Fu H.: Interactive sketch-based normal map generation with deep neural networks. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1*, 1 (2018), 22.

Shin H., Igarashi T.: Magic canvas: interactive design of a 3-d scene prototype from freehand sketches. In *Proceedings of Graphics Interface 2007* (2007), ACM, pp. 63–70.

Sỳkora D., Kavan L., Čadík M., Jamriška O., Jacobson A., Whited B., Simmons M., Sorkine-Hornung O.: Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transactions on Graphics (TOG) 33*, 2 (2014), 16.

Schroff F., Kalenichenko D., Philbin J.: Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 815–823.

Sharif Razavian A., Azizpour H., Sullivan J., Carlsson S.: Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014), pp. 806–813.

Simo-Serra E., Iizuka S., Ishikawa H.: Mastering sketching: Adversarial augmentation for structured prediction. *ACM Transactions on Graphics (TOG) 37*, 1 (2018), 11.

Sỳkora D., Sedlacek D., Jinchao S., Dingliana J., Collins S.: Adding depth to cartoons using sparse depth (in) equalities. *Computer Graphics Forum 29* (2010), 615–623.

Simo-Serra E., Trulls E., Ferraz L., Kokkinos I., Fua P., Moreno-Noguer F.: Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 118–126.

Saito T., Takahashi T.: Comprehensible rendering of 3-d shapes. In *ACM SIGGRAPH Computer Graphics 24* (1990), 197–206.

Shi J., Tomasi C.: *Good Features to Track*. Tech. rep., Cornell University, 1993.

Shao T., Xu W., Yin K., Wang J., Zhou K., Guo B.: Discriminative sketch-based 3d model retrieval via robust shape matching. *Computer Graphics Forum 30* (2011), 2011–2020.

Song J., Yu Q., Song Y.-Z., Xiang T., Hospedales T. M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *ICCV* (2017), pp. 5552–5561.

Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Tasse F. P., Dodgson N.: Shape2vec: semantic-based descriptors for 3d shapes, sketches and images. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 208.

Tian Y., Fan B., Wu F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 661–669.

Wang F., Kang L., Li Y.: Sketch-based 3d shape retrieval using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1875–1883.

Wu C.-Y., Manmatha R., Smola A. J., Krahenbuhl P.: Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 2840–2848.

Wagg D. K., Nixon M. S.: On automated model-based extraction and analysis of gait. In *Automatic Face and Gesture Recognition, 2004. Proceedings of Sixth IEEE International Conference on* (2004), IEEE, pp. 11–16.

Whited B., Noris G., Simmons M., Sumner R. W., Gross M., Rossignac J.: Betweenit: An interactive tool for tight inbetweening. *Computer Graphics Forum 29* (2010), 605–614.

Xing J., Chen H.-T., Wei L.-Y.: Autocomplete painting repetitions. *ACM Transactions on Graphics (TOG) 33*, 6 (2014), 172.

Xu P., Fu H., Au O. K.-C., Tai C.-L.: Lazy selection: A scribble-based tool for smart shape elements selection. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 142.

Xing J., Wei L.-Y., Shiratori T., Yatani K.: Autocomplete hand-drawn animations. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 169.

Yi L., Kim V. G., Ceylan D., Shen I.-C., Yan M., Su H., Lu C., Huang Q., Sheffer A., Guibas L.: A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG) 35*, 6 (2016), 1–12.

Yu Q., Liu F., Song Y.-Z., Xiang T., Hospedales T. M., Loy C.-C.: Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 799–807.

Yoon S. M., Scherer M., Schreck T., Kuijper A.: Sketch-based 3d model retrieval using diffusion tensor fields of suggestive contours. In *Proceedings of the 18th ACM International Conference on Multimedia* (2010), ACM, pp. 193–200.

Yan C., Vanderhaeghe D., Gingold Y.: A benchmark for rough sketch cleanup. *ACM Transactions on Graphics 39*, 6 (2020).

Yu Q., Yang Y., Song Y.-Z., Xiang T., Hospedales T.: Sketch-a-net that beats humans. *arXiv preprint arXiv:1501.07873* (2015).

Zhou T., Jae Lee Y., Yu S. X., Efros A. A.: Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1191–1200.

Zagoruyko S., Komodakis N.: Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 4353–4361.

Zhou T., Krahenbuhl P., Aubry M., Huang Q., Efros A. A.: Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 117–126.

Zhang H., Liu S., Zhang C., Ren W., Wang R., Cao X.: Sketchnet: Sketch classification with web images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 1105–1113.

Zhu Z., Rao C., Bai S., Latecki L. J.: Training convolutional neural network from multi-domain contour images for 3d shape retrieval. *Pattern Recognition Letters* (2017).

Zhu F., Xie J., Fang Y.: Learning cross-domain neural networks for sketch-based 3d shape retrieval. In *AAAI* (2016), pp. 3683–3689.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Data S1

Video S1