DanbooRegion: An Illustration Region Dataset

Lvmin Zhang[1,2], Yi JI[2], and Chunping Liu[2]

[1] Style2Paints Research
[2] Soochow University
lvminzhang@acm.org, lvminzhang@siggraph.org, {jiyi,cpliu}@suda.edu.cn

Abstract. Region is a fundamental element of various cartoon animation techniques and artistic painting applications. Achieving satisfactory region is essential to the success of these techniques. Motivated to assist diversiform region-based cartoon applications, we invite artists to annotate regions for in-the-wild cartoon images with several application-oriented goals: (1) To assist image-based cartoon rendering, relighting, and cartoon intrinsic decomposition literature, artists identify object outlines and eliminate lighting-and-shadow boundaries. (2) To assist cartoon inking tools, cartoon structure extraction applications, and cartoon texture processing techniques, artists clean-up texture or deformation patterns and emphasize cartoon structural boundary lines. (3) To assist region-based cartoon digitalization, clip-art vectorization, and animation tracking applications, artists inpaint and reconstruct broken or blurred regions in cartoon images. Given the typicality of these involved applications, this dataset is also likely to be used in other cartoon techniques. We detail the challenges in achieving this dataset and present a human-in-the-loop workflow namely Feasibility-based Assignment Recommendation (FAR) to enable large-scale annotating. The FAR tends to reduce artist trails-and-errors and encourage their enthusiasm during annotating. Finally, we present a dataset that contains a large number of artistic region compositions paired with corresponding cartoon illustrations. We also invite multiple professional artists to assure the quality of each annotation.

Keywords: artistic creation · fine art · cartoon · region processing

1 Introduction

Starting from the composition impression within traditional on-paper drawing crafts, and popularized in fashion strategies beyond modern digital creation workflows, the fundamental art element "region" continuously contributes to the distinctive feeling and unique style beyond multifarious artworks. Nowadays, diversified cartoon processing techniques and commercial cartoon animation workflow facilitate the usage of such important regions.

Aimed at assisting various cartoon image processing techniques, we invite artists to manually annotate regions in cartoon illustrations and digital arts. We observe the demands of real-world cartoon creation workflows and diversiform professional cartoon editing software, and our motivation is based on the demands

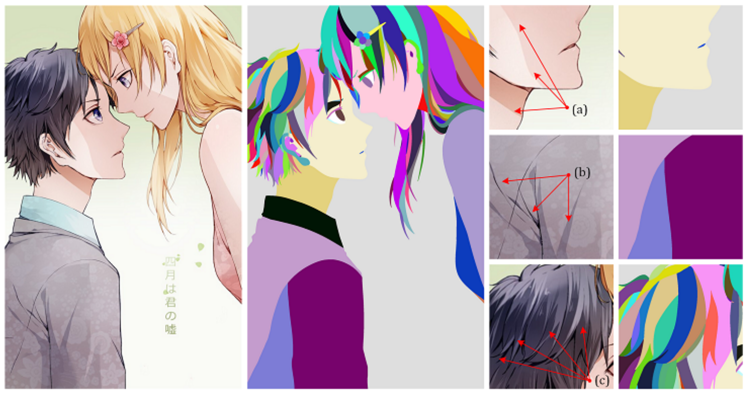


Fig. 1. Example of our region annotating principles created manually by artists.

of related applications. To be specific, we encourage artists to achieve the following application-oriented goals.

Firstly, artists are encouraged [3] to produce object outlines and objects' surface regions by distinguishing and eliminating lighting-and-shadow boundaries. The motivation is to aid in a variety of cartoon rendering/relighting [47, 16] and illumination decomposition [5, 2] applications that require the painted object surfaces to be segmented into independent regions, whereas the edges of light and shadow should not interfere the segmentation. Artists can identify these patterns manually, *e.g.*, in Fig. 1-(a), artists outline the boy face and erase the shadow edge, and these manual data are helpful for algorithms to identify object edges and shadow edges.

Secondly, artists are encouraged to clean up texture or deformation patterns and emphasize cartoon structural lines. The motivation is to help an increasing number of line inking [43, 41, 42] and sketch-based editing [56, 48, 38] tools that are dedicated to achieving cartoon lines faithful to artist perception, plus texture removal [54, 11] and structure extraction [28] scenarios that are aimed at eliminating texture patterns and tones. Artists can determine salient structures and texture patterns in illustrations, *e.g.*, in Fig. 1-(b), artists trace the cloth structure and erase the folding texture, and these data created by artists manually are technically useful.

Thirdly, artists are encouraged to reconstruct, reorganize, and inpaint vague, broken, or missing regions in illustrations. The motivation is to assist a majority of cartoon digitization [27], cartoon topology construction [14, 35], and cartoon or clip-art vectorization [57] workflows, where the cartoon regions are required to be separated as completely as possible, and simultaneously, the closure of these

[3] Although we *encourage* artists to follow these suggestions, they are not absolutely constrained to do so, in order to capture a realistic distribution of artistic region compositions.

regions has to be ensured. Many in-the-wild illustrations have missing, broken, or blurred regions, *e.g.*, in Fig. 1-(c), the ambiguous hair structure is troublesome for vision techniques to interpret, whereas artists can distinguish and manage these regions from human perception.

More importantly, we would like to point out that, given the typicality of the mentioned applications, our dataset is also likely to be used in many other cartoon processing techniques. This is not only because that region is an ubiquitous element in cartoon image processing and this dataset enables many problems to be studied in a data-driven manner, but also because that the availability of the manual data created by real artists is likely to facilitate researches to study the artist perceptual vision in various scenarios, *e.g.*, to study how artists compose the regions when creating their artworks.

To this end, we elaborate the challenges in achieving this dataset, and propose the Feasibility-based Assignment Recommendation (FAR) workflow to enable large-scale annotating. We also present a brief examination where we study the performance of applications using this dataset in many tasks including animation tracking, cartoon intrinsic decomposition, and sketch colorization.

2 Background

Related datasets. Many computer vision or graphics datasets exist in the field of cartoon and creative arts. For example, [61] is a dataset with sketches of semantic segmentations, [21] is a clip-art dataset with class labels, [52] provides artistic images with attributes, [40] is for artistic renderings of 3D models with computer-generated annotations. [36, 3] are existing tools and strategies to collect large datasets of segmented images. [6] discusses how to combine crowd-sourcing and computer vision to reduce the amount of manual annotations. [34, 19, 60] are proposed as applications for these datasets.

Cartoon image segmentation. Previous cartoon image segmentation problems are routinely solved within two paradigms. The first paradigm [57, 17, 12] is to handcraft region priors and solve the regions using traditional algorithms. The second paradigm [28, 20, 55] is to synthesize task-specific datasets and train neural networks with the synthesized data. Our dataset enables the third paradigm: addressing the cartoon image segmentation problem with full access to real data from professional artists in a data-driven manner.

Region-based cartoon application. Various cartoon applications can benefit from our artist annotated regions. Firstly, cartoon image segmentation methods [57] and clip-art vectorizing applications [27] can use a learning-based backend trained on our dataset. Then, manga structure extraction [28], cartoon inking [43, 41, 42], and line closure [29, 30] literatures can also benefit from our region boundaries. Besides, a data-driven region extraction model can be used in cartoon image editing and animating [45]. Moreover, our regions can be used in proxy construction [47, 16] to achieve nonphotorealistic rendering. When applied to animations, our

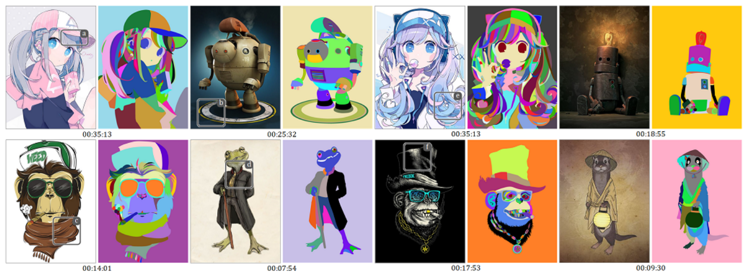


Fig. 2. Selected annotations from professional artists and the corresponding annotating time. We show perceptually satisfying results that are carefully selected from the dataset.

region data can benefit cartoon tracking problems like [59]. Our object boundary annotations can assist cartoon intrinsic decomposition [5, 2]. Finally, artistic region compositions can be applied to line drawing color filling applications [56, 48, 46], or cleaning up the color composition in existing illustrations [18].

Photography segmentation and region evaluation metrics. Although the other "photography" segmentation problems are also extensively studied in non-cartoon image processing works [53, 1, 12, 17, 32, 13, 39], a dominant majority of cartoon processing literatures [30, 43, 41, 28, 42, 29] have reached the consensus that those photography segmentation methods are far from satisfactory when handling cartoons. Nevertheless, the photography region evaluation metrics are still technically rooted and effective when applied to cartoon region datasets. We build our benchmark upon those standard region evaluation metrics and make it consistent to previous region processing literatures.

3 The DanbooRegion Dataset

We introduce a dataset of illustration and region annotation pairs. Specifically, each pair consists of an in-the-wild illustration downloaded from the Danbooru-2018 [15], accompanied by a region map of all pixels marked with a limited number of mutually exclusive indices indicating the structural regions in the original illustration. All samples are provided as RGB images at 1024-px resolution and 8-bit depth, and the region maps are provided as int-32 index images.

The major uniqueness of the presented dataset is that our annotations are carefully created by 12 professional artists. We show several cherry-picked examples in Fig. 2. Our annotations have achieved several results. Firstly, artists produce object outlines and objects' surface regions free from shadow or highlight interference, as shown in Fig. 2-(a,f). Secondly, artists clean up the texture or deformation patterns to obtain structural boundary lines, as shown in Fig. 2-(b,c). Thirdly, artists reconstruct, reorganize, and inpaint the broken or vague regions in illustrations, as shown in Fig. 2-(d,e). As we have mentioned before, these results can benefit various cartoon applications like cartoon intrinsic image, relighting, coloring, inking, shading, vectorizing, tracking, and so on.

We take special care of the annotation quality. Our dataset consists of 5377 region annotation pairs, and the quality of each sample is assured by at least 3 professional artists. In general, 2577 annotations are acknowledged by 5 artists as usable, 1145 annotations are assured by 4 artists, and the remaining 1655 annotations are assured by 3 artists. Furthermore, from the perspective of annotation approach, 154 annotations are painted by artists manually, 146 annotations are collected by aligning and refining publicly available online illustration and line drawing pairs, and the remaining 5077 annotations are achieved using a human-in-the-loop annotation approach. The high-quality region annotations are perceptually satisfying and practically applicable to diversiform downstream tasks related to illustration, cartoon, manga, and painting.

4 Feasibility-based Assignment Recommendation

The objective of our data collection is to gather a large number of illustrations with paired region annotations. Those regions should be either created by artists manually, or coarsely generated by algorithms and then carefully refined by artists. All region annotations must be acknowledged by multiple professional artists as practically usable. Our source illustrations are downloaded from Danbooru-2018 [15] and 12 artists are invited as our annotators. To reduce artist labor, we use a human-in-the-loop workflow: artists create annotations for neural networks to learn, and neural networks estimate coarse annotations for artists to refine.

In order to benefit related applications by achieving our aforementioned goals, we have to face several problems. Firstly, when annotating cartoon object outlines and objects' surface regions, we find many downloaded illustrations are of low quality, and do not have decent and worthwhile object outlines for artists to annotate. Secondly, when cleaning up the texture or deformation patterns and emphasizing cartoon structural lines, we find many sampled images are not related to cartoon, and their texture patterns or structure lines are not suitable for the dataset. Thirdly, when reconstructing, reorganizing, and inpainting the broken or vague regions, we find many illustrations have too many regions, which are impractical and cannot be annotated within an acceptable amount of time. We provide examples in later experiments, where we also show that these problems can disable large-scale annotating because manually solving these problems can significantly waste artists' labor and discourage their enthusiasm.

These problems are non-trivial because of the following reasons. (1) These problems are caused by a large number of intertwined factors, which require huge efforts if engineered one-by-one independently. For example, to identify low-quality or non-cartoon images, the intertwined factors may include color distribution, shape geometry, texture style, semantic feature, and much more. (2) It depends on human perception to determinate whether an illustration can be annotated. For example, there is no fixed threshold to determine how many regions should be considered as impractical to annotate. (3) The involved problem modeling requires assumptions or prior knowledge, making it foreseeably challenging for feature engineering approaches to discriminate illustrations that are not suitable for our dataset.

We present an interactive solution: neural networks learning artist behaviors. Corresponding to the above reasons, our solution has several advantages: (1) It is a joint solution that does not model the intertwined factors independently. (2) It involves human perception. (3) It learns the dataset priors in a data-driven manner. To be specific, we allow artists to give up several types of illustrations during the annotating workflow, and train neural networks to learn artists' giving-up behavior. Firstly, we allow artists to give up low-quality illustrations with no worthwhile object outlines and objects' surface regions. Secondly, we allow artists to give up non-cartoon images without cartoon texture and cartoon inking lines. Thirdly, we allow artists to give up illustrations with too many regions that are impractical to reconstruct, reorganize, or inpaint. Simultaneously, we label these given-up illustrations as "infeasible" and the remaining illustrations as "feasible". We train neural networks with these "feasibility" labels to characterize future illustrations. We name this workflow after Feasibility-based Assignment Recommendation (FAR).

As shown in Fig. 3, the FAR architecture has unique purposes. Firstly, to improve object outline and surface region annotation quality, we need to discard low-quality illustrations as adequately as possible. Therefore, instead of applying a fixed threshold to the CNN output, we rank a batch of illustrations and only view the best one as feasible. Secondly, to aid in cartoon structural region annotating and cartoon texture eliminating, the non-cartoon images should be recognized accurately. Thus, we use the features from different convolutional layers to achieve a reliable multiple-scale estimation. Thirdly, to assist artists to reconstruct broken or vague regions, we need to avoid impractical images with too many regions to annotate. Accordingly, we use the global average pooling to approximate global counting operations in deep convolutional features.

Overview. As shown in Fig. 3, we use a CNN to rank the illustration feasibility and give artists the best one. Then, we record whether that assignment is finished (then labeled as positive) or given up (then labeled as negative), and train the CNN with the recorded labels. In this way, the FAR model is optimized progressively to recommend artists with feasible illustration assignments.

Generating assignment. Each time when an artist queries a new assignment, we randomly sample 100 illustrations in Danbooru and rank their feasibility. Then, we feed the best illustration to the Coarse CNN (Fig. 3) to get a coarse annotation. After that, the illustration and coarse annotation are packaged into an assignment for the awaiting artist. Herein, the Coarse CNN is a U-net [37] trained with the *region-from-normal* approach.

Normal-from-region transform. As shown in Fig. 4, given an input region annotation $\boldsymbol{X}$, the *normal-from-region* transform is aimed at produce a transformed

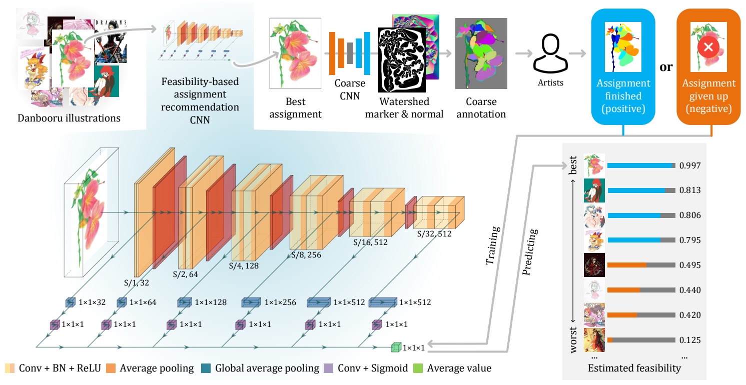


Fig. 3. Feasibility-based Assignment Recommendation (FAR) workflow. We show the neural network architectures and the visualized components within the FAR workflow.

map $\boldsymbol{Y}$, which is a concatenation of a normal map $\boldsymbol{N}$ and a watershed marker $\boldsymbol{W}$. The transform includes the following steps: *Step 1:* Compute the boundary of the region annotation $\boldsymbol{X}$ to get the boundary map $\boldsymbol{B}$. *Step 2:* Compute the Zhang-Sun-Skeleton [58] of the region annotation $\boldsymbol{X}$ to get the skeleton map $\boldsymbol{S}$. *Step 3:* Initialize a random field $\boldsymbol{D_{rand}}$. If one pixel belongs to boundary in $\boldsymbol{B}$, that pixel will be marked as zero. If one pixel belongs to skeleton in $\boldsymbol{S}$, that pixel will be marked as one. The remaining pixels are be marked with random value sampled from random uniform distribution between zero and one. *Step 4:* Optimize the random field $\boldsymbol{D_{rand}}$ to get the displacement map $\boldsymbol{D}$. We uses a routine Gaussian energy

$$\begin{aligned} E_{\text{displacement}} = & \sum_{p} ||(\boldsymbol{D_{rand}})_p - (g(\boldsymbol{D_{rand}}))_p||_1 \\ & + \sum_{i \in \{i | \boldsymbol{B_i}=1\}} ||(\boldsymbol{D_{rand}})_i - 0||_1 \\ & + \sum_{j \in \{j | \boldsymbol{S_j}=1\}} ||(\boldsymbol{D_{rand}})_j - 1||_1 \end{aligned} \tag{1}$$

where p, i, and j are possible pixel positions, $g(\cdot)$ is a Gaussian (sigma is 1.0) filter, and $|| \cdot ||_1$ is the L1 Euclidean distance. This energy can be flexibly solved by gradient descent. *Step 5:* Compute the normal map $\boldsymbol{N}$ using the displacement map $\boldsymbol{D}$. We use a standard *normal-from-height* [25] algorithm to achieve the normal. *Step 6:* Compute the watershed marker $\boldsymbol{W}$ by binarizing the displacement map $\boldsymbol{D}$. We use the threshold 0.618. *Step 7:* Concatenate the normal map $\boldsymbol{N}$ and the watershed marker $\boldsymbol{W}$ into the final output $\boldsymbol{Y}$.

Region-from-normal transform. Given the concatenated $\boldsymbol{Y}$, we split it into the normal map $\boldsymbol{N}$ and the watershed marker $\boldsymbol{W}$. After that, we run the

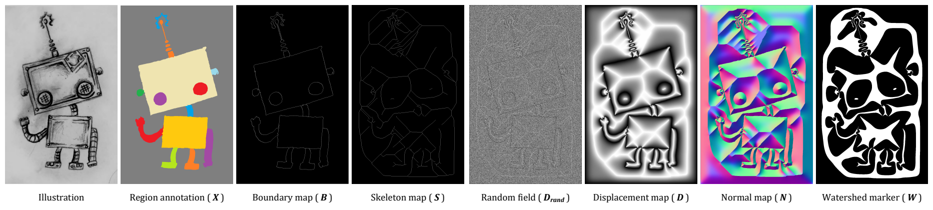


Fig. 4. Visualization of involved maps in the region-normal transform algorithms.

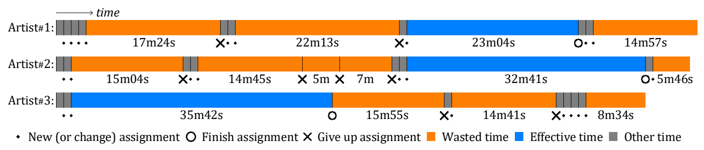
watershed [33] with the marker $\boldsymbol{W}$ filling the map $\boldsymbol{N}$. The results are the reconstructed regions. Note that when the marker $\boldsymbol{W}$ is predicted from neural networks, *e.g.*, the Coarse CNN, we may use morphology methods to remove some possible noise. In particular, we enforce all white regions in $\boldsymbol{W}$ to be bigger than 64 connected pixels.

Handling assignment. Artists are allowed to give up any assignment, and in this case, the assignment feasibility is labeled as "0". Otherwise, the feasibility is labeled as "1", and artists refine the coarse annotations. Firstly, artists refine the region estimation to outline object surface regions free from lighting-and-shadow boundaries. Secondly, artists eliminate texture and emphasize cartoon structural lines in the estimated coarse regions. Thirdly, artists retouch the region estimation to reconstruct, reorganize, and inpaint broken or vague regions in the original illustration.

Instructions given to the artists. We present an artist guidelines in the supplementary material. The artists draw regions following their daily digital painting region composition workflow and the principles in §1. Herein, the digital painting region composition workflow refers to the workflow where artists compose there regions before drawing an illustration or filling colors in an artwork. We provide several on line resources of such workflow in the reference [7, 31].

Initialization. We manually collect 300 low-quality or non-cartoon images as infeasible labels, and then collect another 300 feasible illustrations. In the feasible illustrations, 154 are high-ranking artworks in Danbooru, and 12 artists annotate them manually. The remaining 146 are cartoon images paired with line drawings searched with Google Images, enabling artists to directly align and refine the boundary lines into usable regions.

Detailed workflow. At the beginning, we use the initial 300 feasible labels and 300 infeasible labels to train the FAR ranking network for 25 epochs, and use the 300 initial annotations to train the Coarse CNN for 20 epochs. After that, we view 100 annotations as one loop. When each loop is finished, we train FAR ranking network and Coarse CNN on the new data for 50 epochs, and train them on all old-and-new data for 20 epochs. We use Adam [26] ($l_r = 5e-4$) for



Artist	Wasted time
#1	53m (70.13%)
#2	46m (58.97%)
#3	37m (51.39%)

Fig. 5. Visualization of a naive annotation workflow by simply asking artists to handle all assignments one-by-one. Each time an artist finish or give up an assignment, a new assignment is then automatically assigned to that artist.

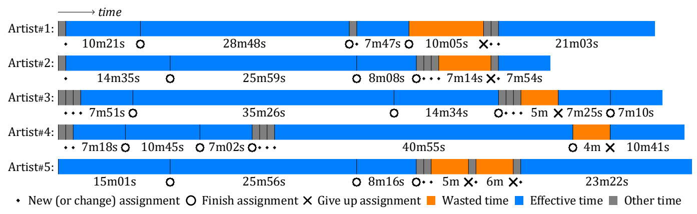
optimization, with each batch containing 8 randomly cropped 256 × 256 samples. Artists refine regions by drawing or erasing region boundaries, and merging or spiting region blocks. These editing strategy can be achieved using the software including PhotoShop, ClipStudio, *etc*.

Quality assurance. After each loop, all 12 artists are invited to check the annotation quality in that loop. Artists are allowed to remove any annotations when they find low-quality ones. This quality assurance stage finishes when each annotation is assured by at least 3 artists as practically usable.

Significance of FAR As shown in Fig. 5, we compare FAR to a naive workflow without the assignment recommendation but directly asking artists to annotate illustrations one-by-one. Artists are also allowed to give up infeasible assignments, and the configuration of the Coarse CNN remains unchanged. Our observation is that the decision of whether to give up is much more difficult than it seems to be, and even professional artists are likely to waste a significant amount of time in trials-and-errors, supported by two evidences: (1) Artists need a period of time to check each illustration assignment before they can judge whether to give up. This checking time causes a huge waste within a large number of assignments, *e.g.*, the artist #1 (Fig. 5) has given up (or change) more than 10 illustrations in the visualized workflow. (2) Even if the artist has decided to cope with one assignment, it is still possible that the assignment will be given up finally. In many cases, only after trying can an artist determine whether the annotating is feasible, *e.g.*, the artist #1 (Fig. 5) have wasted about 70% time (53 minutes) in the workflow. This can cause even worse wastes, discourages artist enthusiasm, and disable large-scale annotating.

On the contrary, by using FAR to recommend artists with feasible assignments, the amount of wasted time can be significantly reduced. We present a recorded workflow as shown in Fig. 6, and we focus on the behaviors of the artist #1. We can see that only 10 minutes are wasted in the beginning 79 minutes. The success of such effective workflow comes from the fact that many infeasible assignments are discriminated by FAR, and the consecutive success also protect artist enthusiasm and keep them encouraged.

Furthermore, we provide examples of infeasible illustrations discriminated by FAR in Fig. 7, where we also visualize the heat-map computed with Grad-Cam++ [8]. For example, as shown in Fig. 7-(b), several illustrations have too



Artist	Wasted time
#1	10m (13.15%)
#2	7m (11.47%)
#3	5m (6.67%)
#4	4m (5.13%)
#5	11m (13.41%)

Fig. 6. Visualization of the annotation workflow using the FAR technique. The statistics are collected at the beginning of each artist's workflow.

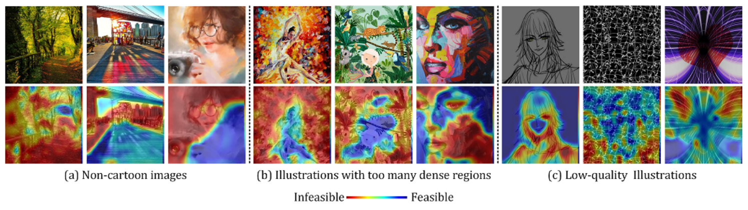

(a) Non-cartoon images
(b) Illustrations with too many dense regions
(c) Low-quality Illustrations
Infeasible
Feasible

Fig. 7. Examples of the infeasible illustrations discriminated by FAR and the Grad-Cam++ heat-map visualization.

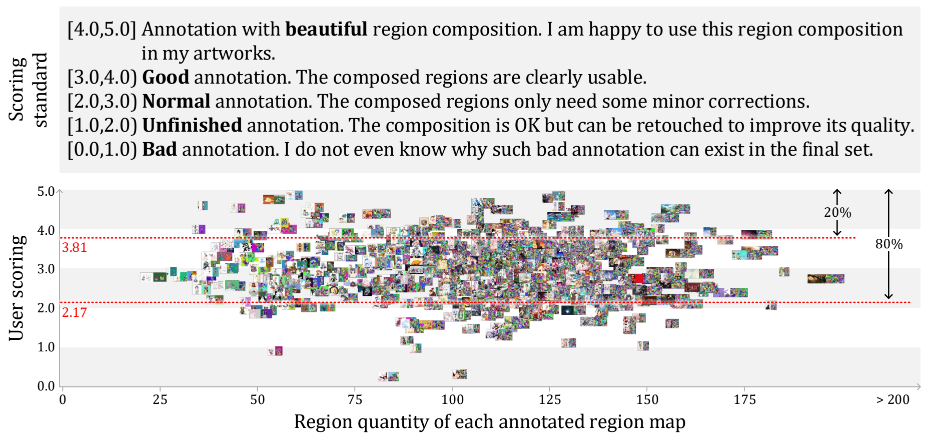
many regions that are impractical to annotate, and the FAR model successfully identifies those dense regions (marked with red in the heat-maps in Fig. 7-(b)). These evidences show that the FAR model has learned functional features to discriminate infeasible assignments.

Analysis of Dataset Quality We perform an user study to analyze the final quality of the presented dataset. We invite the involved 12 artists to score 600 random region maps using the standards in Fig. 8. Each artist score 50 region maps. We also visualize the region quantity in each region map. A vast majority (80%) of the annotations in our dataset is usable or only need minor corrections, and about 20% annotations are aesthetically pleasing and can be used in professional artwork production workflows. We also note that the elimination of bad samples is still an unsolved open problem and this dataset still includes many "unfinished" or even "bad" annotations. Besides, results show that most region maps have more than 25 but less than 175 regions.

5 Benchmark

In this section, we first discuss the evaluation metrics for the presented dataset, and then test the performance of several possible candidates. The benchmark is presented in Table 1.

Metric. The metrics Average Precision (AP), Optimal Image Scale (OIS), and Optimal Dataset Scale (ODS) are widely considered as the standard metrics


Scoring standard

[4.0,5.0] Annotation with **beautiful** region composition. I am happy to use this region composition in my artworks.
[3.0,4.0) **Good** annotation. The composed regions are clearly usable.
[2.0,3.0) **Normal** annotation. The composed regions only need some minor corrections.
[1.0,2.0) **Unfinished** annotation. The composition is OK but can be retouched to improve its quality.
[0.0,1.0) **Bad** annotation. I do not even know why such bad annotation can exist in the final set.

Fig. 8. User study result on annotation quality. The x-axis is the quantity of regions in each region map, while the y-axis is the user scoring with the above scoring standard.

in a dominant majority of region processing literature [53, 1, 12, 17, 32, 13, 39]. AP/OIS/ODS are mathematically rooted and particularly effective when evaluating region maps. It is notable that our problem is different from semantic segmentation tasks, *i.e.*, because the region classification labels are not available, so that many well-known metrics (*e.g.*, IoU) are not applicable. The effectiveness of AP/OIS/ODS has been proved by solid foundations including Holistically-nested Edge Detection [53] (HED), NYU-ObjectRegion [32], Berkeley-BSDS500 [1], and so on.

CNN + Unet-decoder[37] + L2 loss			
Encoder	AP ↑	OIS ↑	ODS ↑
VGG16*[44]	0.710	**0.744**	0.731
VGG19*[44]	0.642	0.716	0.647
Resnet56*[22]	**0.722**	0.737	**0.732**
Resnet110*[22]	0.641	0.725	0.692
Densenet121*[23]	0.645	0.663	0.648

Image-to-image translation			
Model	AP ↑	OIS ↑	ODS ↑
Pix2Pix*[24]	0.562	0.663	0.642
CRN*[9]	0.546	0.606	0.548
Pix2PixHD*[51]	0.568	0.667	0.57

Traditional algorithm			
Algorithm	AP ↑	OIS ↑	ODS ↑
Mean Shift [12]	0.518	0.592	0.524
NCuts [13]	0.428	0.547	0.515
Felz-Hutt [17]	0.404	0.517	0.466
SWA [39]	0.377	0.568	0.538
Quad-Tree [1]	0.17	0.247	0.219
gPb-owt-ucm [1]	0.633	0.651	0.640
CartoonVec [57]	0.536	0.614	0.614

Dense instance segmentation			
Method	AP ↑	OIS ↑	ODS ↑
TensorMask [10]	0.454	0.519	0.495

Table 1. Benchmark of region boundary precision. Scores are in orange if below 0.65 and in blue if above 0.65. Best scores are marked in bold with red background.
↑ refers to higher being better.
* refers to *region-from-normal* approaches.

We include a brief review on AP/OIS/ODS to aid in reproducibility. Given a ground truth region map and an estimated one, AP is the average proportion of the true-positive pixels (correct region boundary pixels) in all positive pixels (all estimated region boundary pixels). We resize region maps to 256px when computing AP. It is obvious that AP is sensitive to image scale. Even computed on same region maps, different resizing scale causes remarkable indeterminacy. To address this problem, OIS searches all possible scales for each independent sample and report the optimal precision, whereas ODS searches all possible scales for the entire test set and report the optimal precision. Moreover, ODS is routinely considered as the best metric by many typical region processing works including HED [53], BSDS500 [1], and so on.

Experimental setting. In the 5377 annotations, the quality of 2577 annotations are assured by 5 professional artists. In order to capture an accurate distribution of artist perception beyond artwork regions, we use the first 1000 in those 2577 annotations as our test set, with the remaining 4377 annotations being the training set. As to data augmentation, we use random left-and-right flipping and random rotation. We then randomly crop image data into 256×256 samples during the online training. It is notable that all non-deep-learning methods are directly tested on our test set.

Algorithm candidate. We test the performance of traditional segmentation algorithms [12, 13, 17, 39, 1], including Vectorizing Cartoon Animations (CartoonVec) [57], by directly applying them to the test set. Then, using the aforementioned *region-from-normal* approach, we test the performance of image-to-image translation methods [24, 51, 9] by training them to predict region normals and markers. In the same way, we also include the baseline performance of VGG [44], Resnet [22], and Densenet [23], by using them as the encoder of U-net [37], and we train them with a standard L2 loss on region normals and markers. Instance segmentation methods can also be applied to our problem. To the best of our knowledge, TensorMask [10] is the only instance segmentation method that does not require region classification labels and can be trained on arbitrary number of regions. Therefore, we also train TensorMask on our dataset and report its performance.

Human performance. Different artists may produce slightly different region annotations even for one same illustration. Therefore, we also invite the 12 involved artists to annotate 36 random test illustrations again (each artist with 3 illustrations), and report the human performance: 0.785(OIS), 0.782(ODS).

Result. We have several interesting discoveries in this benchmark: (1) Learning-based models generally report higher scores than traditional algorithms. This indicates that data-driven approaches seem to be more faithful to artist perception when generating the regions. (2) Shallow models like VGG16 and Resnet56 outperform deep models like VGG19 and Resnet110. This may because deep models tend to over-fit the dataset, and stronger data augmentation might be necessary. (3) The instance segmentation method, TensorMask, does not report impressive performances. This may because the regions in our dataset is denser than most instance segmentation datasets, and instance segmentation methods like TensorMask are not well-suited for our problem.

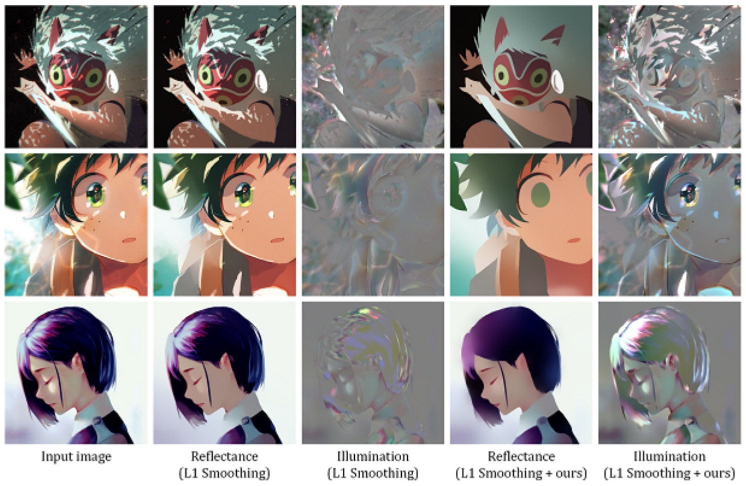

Input image
Reflectance (L1 Smoothing)
Illumination (L1 Smoothing)
Reflectance (L1 Smoothing + ours)
Illumination (L1 Smoothing + ours)

Fig. 9. Intrinsic cartoon images. We show the reflectance maps and the illumination maps decomposed using L1 smoothing with naive regions in [5] and our regions.

6 Application

Cartoon intrinsic images. We present a region-based L1 intrinsic decomposition [5] as in Fig. 9. We can see that our data-driven region segmentation enables L1 smoothing to achieve more adequate decomposition when compared to the naive L1 smoothing. In particular, we directly train a Pix2PixHD [51] for this application. After the training, the estimated normal maps and watershed marker maps can be translated to regions using the *normal-from-region* approach. One notice is that we use a special data augmentation method to augment the luminance domain of the images to avoid luminance over-fitting, by converting the RGB image into Lab image, and then randomly reduce the contrast of the L channel. In particular, we reduce the L contrast by random scalar $U(0.1, 0.9)$, and then translate the Lab image back to RGB image. In order to make the estimated region boundary a bit more smoother (regions produced by watershed is not very smooth in most cases), we use the method [4] to simplify the topology of our region boundary. This is achieved by translating the region map into a vectorized map and then rasterize it back. We carefully tune the parameters of [4] and the smoothing weights to achieve our results in Fig. 9.

Flat sketch colorization and color cleaning-up. Our dataset enables learning-based sketch colorization solutions [56, 48] to be "cleaned-up" so as to adapt to the flat "cel-colorization" workflows like LazyBrush [46], by enabling neural networks to learn to reconstruct broken or vague regions. We present flat colorization and cleaning-up examples as shown in Fig. 10. We train a Pix2PixHD [51] for the application, and the estimation can be translated to regions using the

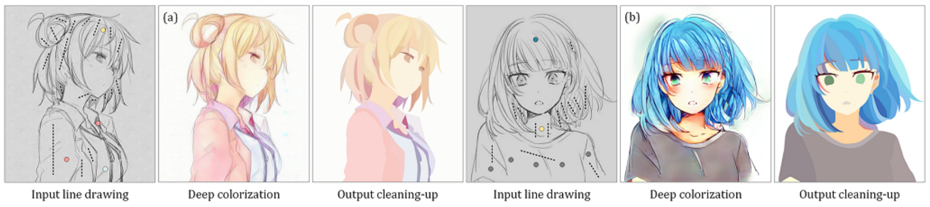


Fig. 10. Sketch colorization cleaning-up. *The regions in this figure are achieved interactively by user.* We use the estimated regions to clean up the sketch colorization results by sampling median color in each region. (a) Colored with PaintsChainer [48]. (b) Colored with Style2Paints [56]. The pointed hints are color indications for interactive coloring, while the dotted hints indicates that its covered regions should be merged.

normal-from-region approach. For data augmentation, we use the high-frequency augmentation. This is to encourage the neural networks to learn to reconstruct broken or ambiguous regions, and avoid the neural networks to over-fit the high-frequency edges in the input image. We apply a Bilateral Filter [49] to the training input image with a random number in $U(20, 1000)$ for the Bilateral spacial sigma, and $U(20, 1000)$ for the Bilateral color sigma. We also allow users to merge some regions in the region map by drawing some "dotted" lines. We use a fine-tuned selected search [50] to merge some small regions. The regions are also processed with [4] to simplify the topology of region boundary and make the resulting region looks smooth, by translating the region map into a vectorized map and rasterize it back. The smoothing parameters are tuned for Fig. 10.

7 Conclusion

We presented a dataset of illustration and region composition pairs annotated by real-life artists. Our dataset is unique in that it is faithful to artist perception, and is costumed to benefit diversiform cartoon processing applications. All annotations are created from in-the-wild cartoon illustrations, and the quality of each annotation is assured by multiple artists. We provide the details of our data collection pipeline, which leverages a novel human-in-the-loop annotation workflow, namely Feasibility-based Assignment Recommendation (FAR), that is designed to improve the feasibility of involved assignments and enable the large-scale annotating. We demonstrate the usage of our dataset in a variety of applications like cartoon tracking, cartoon intrinsic images, and sketch colorization. Finally, we provide considerations for further researches in related avenue.

References

1. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5), 898–916 (may 2011)

2. Bell, S., Bala, K., Snavely, N.: Intrinsic images in the wild. ACM Trans. on Graphics (SIGGRAPH) **33**(4) (2014)
3. Bell, S., Upchurch, P., Snavely, N., Bala, K.: OpenSurfaces: A richly annotated catalog of surface appearance. ACM Trans. on Graphics (SIGGRAPH) **32**(4) (2013)
4. Bessmeltsev, M., Solomon, J.: Vectorization of line drawings via polyvector fields. ACM Transactions on Graphics **38**(1), 1–12 (jan 2019). https://doi.org/10.1145/3202661
5. Bi, S., Han, X., Yu, Y.: An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. ACM Trans. Graph. **34**(4) (Jul 2015). https://doi.org/10.1145/2766946, https://doi.org/10.1145/2766946
6. Branson, S., Van Horn, G., Perona, P.: Lean crowdsourcing: Combining humans and machines in an online system. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
7. caydett: Cel shading tutorial by caydett. https://www.deviantart.com/caydett/art/Cel-Shading-Tutorial-270935090 (2018)
8. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Gradcam: Generalized gradient-based visual explanations for deep convolutional networks. In: WACV. IEEE (mar 2018). https://doi.org/10.1109/wacv.2018.00097
9. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV (2017)
10. Chen, X., Girshick, R., He, K., Dollar, P.: Tensormask: A foundation for dense object segmentation. In Arxiv (2019)
11. Cho, H., Lee, H., Kang, H., Lee, S.: Bilateral texture filtering. ACM Transactions on Graphics **33**(4), 1–8 (jul 2014). https://doi.org/10.1145/2601097.2601188
12. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5), 603–619 (may 2002). https://doi.org/10.1109/34.1000236
13. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: CVPR. IEEE (2005). https://doi.org/10.1109/cvpr.2005.332
14. Dalstein, B., Ronfard, R., van de Panne, M.: Vector graphics animation with time-varying topology. ACM Trans. Graph. **34**(4) (Jul 2015). https://doi.org/10.1145/2766913
15. DanbooruCommunity: Danbooru2017: A large-scale crowdsourced and tagged anime illustration dataset (2018)
16. Dvorožňák, M., Nejad, S.S., Jamriška, O., Jacobson, A., Kavan, L., Sýkora, D.: Seamless reconstruction of part-based high-relief models from hand-drawn images. In: Proceedings of International Symposium on Sketch-Based Interfaces and Modeling (2018)
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)
18. Fourey, S., Tschumperle, D., Revoy, D.: A fast and efficient semi-guided algorithm for flat coloring line-arts. EUROGRAPHICS (2018)
19. Gao, C., Liu, Q., Xu, Q., Wang, L., Liu, J., Zou, C.: Sketchycoco: Image generation from freehand scene sketches. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
20. Gao, S., Janowicz, K., Montello, D.R., Hu, Y., Yang, J.A., McKenzie, G., Ju, Y., Gong, L., Adams, B., Yan, B.: A data-synthesis-driven method for detecting and extracting vague cognitive regions. International Journal of Geographical Information Science pp. 1–27 (jan 2017). https://doi.org/10.1080/13658816.2016.1273357
21. Garces, E., Agarwala, A., Gutierrez, D., Hertzmann, A.: A similarity measure for illustration style. ACM Transactions on Graphics (SIGGRAPH 2014) **33**(4) (2014)

22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (jun 2016). https://doi.org/10.1109/cvpr.2016.90
23. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (jul 2017). https://doi.org/10.1109/cvpr.2017.243
24. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)
25. Kender, J.R., Smith, E.M.: Shape from Darkness: Deriving Surface Information from Dynamic Shadows, chap. 3, pp. 378–385. Jones and Bartlett Publishers, Inc., USA (1992)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. Computer Science (2014)
27. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. ACM Transactions on Graphics (TOG) **26**(3), 3 (jul 2007). https://doi.org/10.1145/1276377.1276381
28. Li, C., Liu, X., Wong, T.T.: Deep extraction of manga structural lines. ACM Transactions on Graphics **36**(4) (2017)
29. Liu, C., Rosales, E., Sheffer, A.: Strokeaggregator: Consolidating raw sketches into artist-intended curve drawings. ACM Transactions on Graphics (2018)
30. Liu, X., Wong, T.T., Heng, P.A.: Closure-aware sketch simplification. ACM Transactions on Graphics **34**(6), 168:1–168:10 (November 2015)
31. MicahBuzan: Cel shading tutorial. https://www.micahbuzan.com/cel-shading-tutorial/ (2020)
32. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
33. Neubert, P., Protzel, P.: Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. ICPR (2014)
34. Ren, H., Li, J., Gao, N.: Two-stage sketch colorization with color parsing. IEEE Access **8**, 44599–44610 (2020)
35. Rivière, M., Okabe, M.: Extraction of a cartoon topology. In: ACM SIGGRAPH 2014 Posters on - SIGGRAPH 14. ACM Press (2014). https://doi.org/10.1145/2614217.2614260
36. Ronchi, M.R., Perona, P.: Describing common human visual actions in images. In: Proceedings of the British Machine Vision Conference (BMVC). pp. 52.1–52.12. BMVA Press (September 2015). https://doi.org/10.5244/C.29.52, https://dx.doi.org/10.5244/C.29.52
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. MICCAI (2015)
38. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. CVPR (2017)
39. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual scenes. Nature **442**(7104), 810–813 (jun 2006). https://doi.org/10.1038/nature04977
40. Shugrina, M., Liang, Z., Kar, A., Li, J., Singh, A., Singh, K., Fidler, S.: Creative flow+ dataset. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
41. Simo-Serra, E., Iizuka, S., Ishikawa, H.: Mastering Sketching: Adversarial Augmentation for Structured Prediction. ACM Transactions on Graphics **37**(1) (2018)
42. Simo-Serra, E., Iizuka, S., Ishikawa, H.: Real-time data-driven interactive rough sketch inking. ACM Transactions on Graphics (2018)

43. Simo-Serra, E., Iizuka, S., Sasaki, K., Ishikawa, H.: Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup. ACM Transactions on Graphics **35**(4) (2016)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. TPAMI (2014)
45. Sýkora, D., Buriánek, J., Žára, J.: Sketching cartoons by example. In: Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling. pp. 27–34 (2005)
46. Sykora, D., Dingliana, J., Collins, S.: LazyBrush: Flexible painting tool for hand-drawn cartoons. Computer Graphics Forum **28**(2) (2009)
47. Sýkora, 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 Transaction on Graphics **33**(2), 16 (2014)
48. TaiZan: Paintschainer tanpopo. PreferredNetwork (2016)
49. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). Narosa Publishing House (1998). https://doi.org/10.1109/iccv.1998.710815
50. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV (2013)
51. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. CVPR (2018)
52. Wilber, M.J., Fang, C., Jin, H., Hertzmann, A., Collomosse, J., Belongie, S.: Bam! the behance artistic media dataset for recognition beyond photography. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
53. Xie, S., Tu, Z.: Holistically-nested edge detection. In: CVPR (2015)
54. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Transactions on Graphics **31**(6), 1 (nov 2012). https://doi.org/10.1145/2366145.2366158
55. Xu, N., Price, B., Cohen, S., Huang, T.: Deep image matting. In: CVPR. IEEE (jul 2017). https://doi.org/10.1109/cvpr.2017.41
56. Zhang, L., Li, C., Wong, T.T., Ji, Y., Liu, C.: Two-stage sketch colorization. In: ACM Transactions on Graphics (2018)
57. Zhang, S.H., Chen, T., Zhang, Y.F., Hu, S.M., Martin, R.R.: Vectorizing cartoon animations. TVCG (2009)
58. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. Communications of the ACM (1984)
59. Zhu, H., Liu, X., Wong, T.T., Heng, P.A.: Globally optimal toon tracking. ACM Transactions on Graphics **35**(4), 75:1–75:10 (July 2016)
60. Zou, C., Mo, H., Gao, C., Du, R., Fu, H.: Language-based colorization of scene sketches. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019) **38**(6), 233:1–233:16 (2019)
61. Zou, C., Yu, Q., Du, R., Mo, H., Song, Y.Z., Xiang, T., Gao, C., Chen, B., Zhang, H.: Sketchyscene: Richly-annotated scene sketches. In: ECCV. pp. 438–454 (2018)