EvaToon: A Novel Graph Matching System for Evaluating Cartoon Drawings

Yirui Wu†, Xianli Zhou†, Tong Lu†*, Guo Mei‡, Linbi Sun‡
†National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China
‡Nanjing Technical Vocational College, Nanjing, China
Email:wuyirui1989@163.com; 2316150050@qq.com; lutong@nju.edu.cn; 875228106@qq.com; 45495757@qq.com

***Abstract*—Imitation cartoon drawing is an important skill for cartoonists, requiring quantity of efforts on practising and guidance. In this paper, we propose EvaToon, an imitated drawing evaluate system, which automatically assigns judging scores and marks improper drawing regions. With our system, cartoonists can practise and get guidance by themselves. We have cooperated with several experts on developing such an evaluation system. Based on their guide, we present EvaToon in two stages comprising *cartoon drawings analyzing* and *similarity evaluating*. During analyzing, we first locate contour pixels with high curvature as interest points and then extract multi-scale features around interest points to hierarchically describe shape. During evaluating, we first match interest points between original and imitated drawing based on distance of features. After matching, we construct a regression tree to map high dimensional difference of matching features to scores and marks based on quantity of manually evaluated training examples. Finally, our system matches an input imitated drawing with the original one and predicts its scores automatically. We demonstrate the accuracy of our EvaToon system in matching and predicting and prove the capability of describing shape of our proposed features by experiments on a collected dataset of imitated drawings.**

I. Introduction

Nowadays there are a lot people love to watch cartoons, which are drawn by cartoonists to represent either humorous or satirical ideas. Cartoon production has been benefited from the recent technologies such as modeling [1], rendering [2] and animation [3]. However, drawing cartoons frame by frame is still one of the most skilled task. This is true because generally only those well-trained cartoonists can do it well. Therefore, there is a new demand in the document analysis and graphics recognition community to provide efficient tools for training inexperienced cartoonists.

The conventional cartoon training process consists of the following two stages, namely, an inexperience cartoonist first imitates existing cartoon drawings, and then experts score his/her imitation artworks by marking unappropriate regions on their cartoon drawings for improvement. If the scoring process can be automatically performed with real-time and accurate feedbacks for corrections but need not wait for expert feedbacks, the cartoonist can improve his/her drawing skills by a more efficient way (see Fig 1). However, automatically evaluating cartoon drawings is challenging due to its inherent difficulty in characterizing artistic cartoon drawings quantitatively. In the past year, we have cooperated with several cartoon experts on developing new tools for evaluating

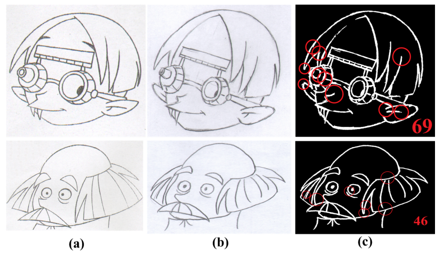

Fig. 1. The idea for the proposed evaluating system that automatically evaluates an imitation cartoon drawing: (a) the original cartoon drawings, (b) the imitation cartoon drawings, and (c) scores and marks generated by our EvaToon system.

imitation cartoon drawings. During our cooperations, we find the most important criterion for cartoon evaluating relies on providing shape evaluation tools between the original and imitated cartoon drawings. In other words, a well designed cartoon evaluating system need first describe various cartoon shapes properly and then match these cartoons for scoring. Thereby, the main challenges for cartoon evaluation lie in 1) how to represent cartoon shape features, and 2) how to involve expert knowledge for scoring a cartoon drawing.

In this paper, we propose a novel and efficient cartoon drawing evaluating system, which we call EvaToon, to automatically evaluate the imitation of cartoon drawings, including scoring drawings and marking improper drawing regions. Note that we build our evaluating system on the supposition that imitated drawings are similar in the overall appearances, which coincides with the scenario of practising for inexperienced cartoonists. We represent such an evaluating system with two stages, namely, shape analysis and shape evaluation, to meet the above challenges.

During the shape analysis stage, we represent the shape of a cartoon drawing with two structures: image patches and lines. Image patches divide the drawing to local parts. Since the details of the shape vary in patches, cartoonists need great efforts to achieve a desirable local shape, including orientation, spatial relation and so on. Although patches can represent the shape to some extent, noises in patches will still lead to misunderstanding of drawing content. These noises may be caused by shakes and breaks of lines or undesirable scan quality. Therefore, we propose to utilize lines to improve the robustness of describing a shape. Our proposed line-

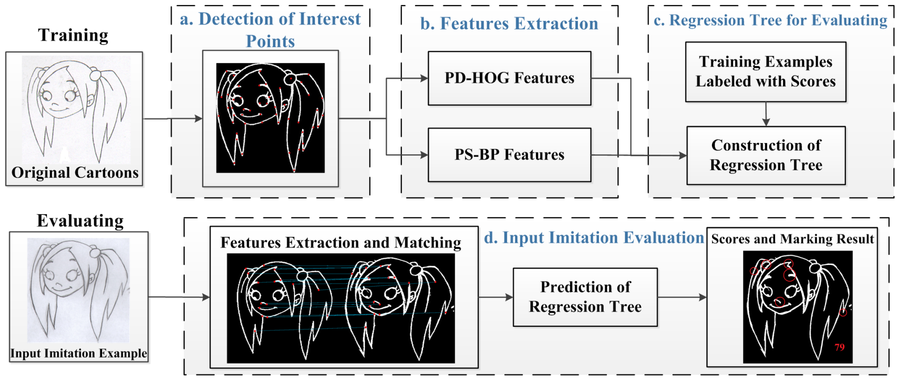

Fig. 2. The proposed framework for evaluating cartoon drawings: (a) detection of interest points from each original cartoon drawing, (b) extracting features around interested points, (c) constructing a regression tree for evaluation, and (d) evaluating the input cartoon imitated drawing.

based features offer flexibility to discard noises and focus on lines, which are essentially the fundamental parts of cartoon drawings. Specifically, we first locate interest points by curvature analysis, and then extract shape-aware and local features around interested points. One of the features, named as PD-HOG, describes the local shape with patches, while the other one named as PS-BP encodes the local patterns of lines.

In the shape evaluation stage, we involve the criterion from experts to evaluate each cartoon drawing with a supervised learning scheme. By gathering a quantity of evaluation examples, we construct a binary regression tree to map high dimensional features to corresponding evaluating scores. Here, we utilize a regression tree for mapping due to its capability of imitating the judgment from experts. During judging, experts gradually assign scores based on the classifications of key regions, which represent how imitated drawings are similar to the original one on maintaining the consistence of appearance. Correspondingly, the constructed regression tree classifies the examples in non-leaf nodes based on similarity calculation between multi-scale features of matching interested points in a more global sense, and then predicts a convinced score and the number of markings in leaf nodes.

The main contribution of the paper is to propose a platform to evaluate cartoon drawings, which also supports scoring and marking improper regions. We propose two new features to describe cartoon shapes. The features well encode cartoon shape characteristics, which improve the robustness and completeness for describing cartoon shapes. Expert knowledge for evaluation is also integrated in the proposed platform by a learning scheme. We show that the proposed EvaToon platform can well relieve the burden of training cartoonists.

II. Related Work

To our best knowledge, there are only a few works related to the evaluation of cartoon drawings in the past decades. The existing methods related to our work can be categorized into the following two types: cartoon creation systems and cartoon matching methods.

Cartoon Creation Systems. There are two categories of cartoon creation technologies. One helps to create cartoon characters from real-life images, and the other one guides inexperienced users to draw cartoons. For example, Li et al. [4] propose to guide face cartoon synthesis by constructing a local linear model constrained by the content of images from the training set. However, it requires realistic images as the input, which limits the imagination of cartoonists. Several other systems are designed to assist drawing cartoons by displaying the guidance on drawing surface. The work proposed by Fu et al. [5] maps the Gestalt rules to computational procedures, which generate human-like drawing animations for a given line drawing. With the development of interactive technologies, further attempts have been made to provide assistances. For example, the drawing assistant system in [6] presents an interactive drawing tool, which provides automated guidance over photographs to help people practice drawing-by-observation techniques. However, evaluations are not discussed in the above systems.

Cartoon Matching Methods. Searching the correspondence between cartoon drawings is the key problem of animation production. The existing methods for solving the matching problem can be roughly categorized into two classes: graph-cut based methods [7] and descriptor based methods. Graph-cut methods first construct a specialized graph for the given cartoon image, and then minimize the energy function of the graph. For example, Yu et al. [8] propose a semi-supervised graph model to align local patches in feature space. The other kind of the methods represents a cartoon drawing through appropriate descriptors. Recently, Jin et al. [9] combine simple features (e.g., gray value histogram, the count of corners, and so on) and a predefined 3D face model to achieve reliable correspondence results of multiview cartoon drawings even with occlusions. Since these features are designed to match similar frames of one animation, they are not suitable to match the drawings from inexperienced cartoonists due to their variances of visual appearance.

III. The Proposed Platform

In this section, we propose a novel cartoon drawing evaluation platform to score cartoon drawings with marked improper regions. Fig. 2 gives the overview of the proposed platform, where (a) and (b) compose the analyzing stage, (c) and (d) represent the evaluating stage. Note that we use PD-HOG and PS-BP features in (c) and (d). However, the extracted features in these two procedures are on different scales. Therefore, we utilize their capability of describing shapes on multi-levels to match local areas and measure similarity in the global sense, respectively. Before processing, we covert the scanned cartoon drawing image to a binary PNG file to decrease the influences brought by scanning or noise. After that, we remove noise connected components with small areas, do the dilate operation to discard noise pixels near the drawing lines and segment the result binary image to the drawing part. Our further processing will be performed on the segmented binary image.

A. Detection of Interest Points

Cartoon experts generally evaluate an imitation cartoon drawing based on different key points, which inspires us to consider interested points with abundant neighboring information as the key points. We find points with high curvature is important for describing the appearance of a cartoon drawing. These points often appear as endings, joints and the middle point of a curve, describing local shape characteristics with their neighboring regions. Therefore, we start our evaluation from detecting our interested points that have high curvature.

Inspired by [10], we use the first order derivative of Gaussian to achieve convinced gradient estimation for the input binary cartoon image:

$$g_{i,x} = -\frac{x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

where i represents a pixel in binary image I, and σ is the standard deviation of Gaussian window, x and y refer to the x-coordinate and y-coordinate of pixel i, respectively. Note that the size of Gaussian window is adaptively defined based on σ. We get gradient estimation $g_{i,y}$ in y-direction with Equ.1 as well. For discrete form, Equ.1 is represented as a differential operation on the gaussian-filtered binary image.

We then search for interested points among the set of contour pixels Γ, which share the property of significant gradient values. Specifically, we select counter pixel i as an interested point with its gradient orientation larger than threshold α:

$$U = \{i | o_i > \alpha, \forall i \in \Gamma\} \tag{2}$$

where $o_i \in [0, \pi)$ is the gradient orientation of counter pixel i defined as:

$$o_i = \arccos \frac{g_p \cdot g_q}{|g_p| * |g_q|}, where\ \{p, q\} \in \Gamma \wedge d_{ip} = d_{iq} \tag{3}$$

where p and q are two nearby counter pixels of i constrained by a predefined and small distance value, respectively.

We show the set of original interested points U in Fig. 3(a), where we can see the interested points with high curvature appear in key places to characterize the shape of a cartoon drawing. Since U consists of a group of neighbor points and the size of U is too large for efficiently matching (usually up to 300), we reduce the size of U to a smaller value by

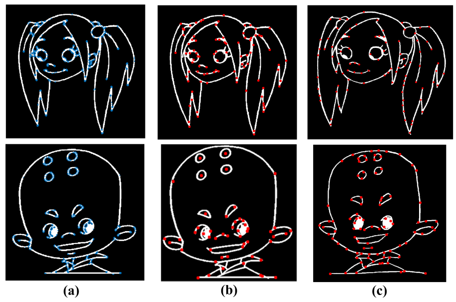

Fig. 3. Examples of detected interested points. (a) the original interested points with high curvature, (b) the interested points for original drawing, (c) the interest points for imitation drawing.

hierarchical clustering, which iteratively emerges the nearest points and constructs a cluster tree from bottom to top. We adaptively decide the number of clusters n_i (the number of interested points) by maximizing the silhouette value, which generally keeps a sufficient number of interested points in to remain the balance between representing a cartoon shape and efficient matching. The center of each cluster forms the result of interested points as shown in Fig. 3(b)-(c). Note that we assign an interested point in $\tilde{U}$ far away from lines to its nearest one in U to avoid sparse features.

B. Features Extraction and Matching

In this subsection, we aim at constructing multi-scale features to encode shape information of a cartoon drawing, and further match interested points each imitated drawing with the original drawing based on constructed features. Specifically, we propose two novel features, named as PD-HOG and PS-BP, to describe the shape around the interested points. PD-HOG gathers gradients orientation information from patches with different sizes to encode a shape hierarchically, while PS-BP focuses on patterns of lines. Note that we address the problem of multi-scale capability in both features since we plan to match local regions while measure the similarity on a global sense. We judge the similarity globally because scoring an imitated drawing not only requires analyzing local regions but also needs the resemblance of the global representations of two different drawings.

PD-HOG is short for **P**yramid **H**istogram **O**f **G**radients for binary **D**rawings. Note that PD-HOG is similar with the HOG descriptor, which extracts gradient-aware features from pixels in patches to describe local shapes. However, our feature is different in several aspects. For HOG, features should be densely sampled to describe appearances from patches in gray images, and they are sensitive to scale changes. In contrast, we design PD-HOG to hierarchically extract the features around interested points to describe the shape of a binary image, which can be formulated as follows:

$$h_{i,l} = f_h(\mathcal{P}_i, G_{I_l}),\ 1 \le i \le n_i \tag{4}$$

where l is the scale value, $\mathcal{P}_i$ refers to the patch centered at the ith interested point in set E, G_{I_l} utilizes Equ.1 to compute the

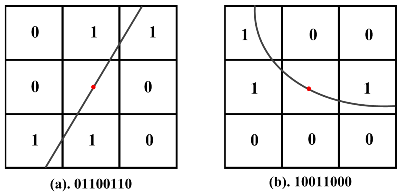

Fig. 4. The examples of our proposed PS-BP feature. The red points are the extracted interested points, while the binary values below are the corresponding feature descriptors for drawing lines.

gradient map of the scaled binary image I_l achieved by down-sampling with scale l, and function $f_h()$ computes histograms of the orientations of gradient vectors inside the patch, which is defined as a block with $16 * 16$ size. The result PD-HOG feature h for the ith interested point can thus be represented as a combination of multi-scale features with an alternative scale number β, i.e. $h_{i,\beta} = |\bigcup_{l=1}^{\beta} h_{i,l}|$, where "| |" refers to $L1$ normalization. Since contour regions own more significant gradient values, PD-HOG feature mainly describes the shape information on contour, which is appropriate for scoring a cartoon drawing.

We propose PS-BP to describe local patterns of the lines constrained by the interested points. PS-BP is short for **P**yramid **B**inary **P**attern of **S**keleton. Inspired by LBP (Local Binary Pattern) [11], we convert the trends of lines to local patterns around the interested points, which can be written as follows:

$$b_{i,l} = f_b(i, f_s(I_l)),\ 1 \leq i \leq n_i \tag{5}$$

where function $f_s()$ refers to the stable skeletonize method [12], while function $f_b()$ represents the process of constructing binary codes for local patterns of lines, which are constrained by the ith interested point. Fig. 4 shows some examples of the proposed PS-BP feature. Specifically, we first assign interested points to its nearest point on the skeleton of lines. Then, the neighboring regions of the interested points are split to eight blocks with edge length γ. Note that the block size used here is nearly the same with that of the block used by PD-HOG. The binary value for a block will be assigned for "1" if the block is passed by drawing lines, otherwise it will be set to "0". We record the binary values clockwise starting from the top-left corner, which results in PS-BP features $b_{i,l}$. After combining, we obtain the final PS-BP feature $b_{i,\beta} = |\bigcup_{l=1}^{\beta} b_{i,l}|$. Note that we discard noises of non-line pixels and concentrate on patterns of thinned lines in PS-BP features, which improves the robustness of representing cartoon shapes.

After constructing the proposed features, we resize the imitated drawing to the same size of the original one, and define $L2$ distance $d_{i,j,\beta}$ between the features based on a linear weight scheme:

$$d_{i,j,\beta} = (1 - \omega) * \|h_{i,\beta} - h_{j,\beta}\| + \omega * \|b_{i,\beta} - b_{j,\beta}\| \tag{6}$$

where ω is an empirically determined weight, i and j are the index of interested points in the original and the imitated drawing, respectively. Finally, we select pairs of interested points with the minimal distance of the features by adopting the Best-First-Bin(BFB) algorithm [13]. Note that we constrain the matching to be unique mapping, i.e. the ith and the jth interest points are matched only if $d_{i,j}$ is the minimal among all candidates for both i and j. We thus construct the matching set $\mathcal{M} = \{(i,j), 1 \leq i \leq n_{i,I_o}\}$, where n_{i,I_o} refers to the number of interested points in the original drawing. Due to unique matching, we may get some missing values of j for the ith interested point in $\mathcal{M}$. Besides, there exist some false matchings due to the resemblance of local shapes, most of which can be eliminated by the Manhattan distance as follows:

$$\tilde{\mathcal{M}} = \{(i,j), if\ |i_x - j_x| + |i_y - j_y| < \varphi * (r + c)\} \tag{7}$$

where r and c are the height and width of original drawing, respectively, and φ is a predefined threshold. We show several matching examples in Fig. 5, where we can notice the completeness and robustness of matching even there exist variances in appearance.

C. Constructing Regression Tree for Evaluating

In this subsection, we firstly describe the regression problem, i.e., predicting scores or the number of marks, and then construct a binary regression tree to model the regression problem. Note that we predict the number of marks due to the fact that cartoon experts prefer to mark different numbers of markings based on the similarity of drawings. Adaptively determining the number of marks help avoid any predefined thresholds when marking, which improves the robustness and completeness of our EvaToon platform.

To predict scores for each category of cartoon based on training examples, the input includes labeled variables O and the difference of matched features D. Note that we set scale β as a larger value ($\beta = 5$) when measuring similarity. Let O consist of scores and number of marker places, i.e. $O = \{(s_\tau, n_\tau)|1 \leq \tau \leq n_e\}$, where n_e is the number of training examples, and n_τ refers to the number of marks in the τth example. Let D consist of difference of features, i.e. $D = \{d_{i,\tau}|0 \leq i \leq n_{i,I_o}, 1 \leq \tau \leq n_e\}$, where $d_{i,\tau}$ represents difference of features designed to measure the similarity of matching features as follows:

$$d_{i,\tau} = \begin{cases} (h_{i,\tau} - h_{j,\tau}, b_{i,\tau} - b_{j,\tau}) & j \neq \emptyset \\ \emptyset & j = \emptyset \end{cases} , for\ (i,j) \in \tilde{\mathcal{M}}_\tau \tag{8}$$

From Equ. 8, we notice the difficulty of prediction mainly lies in high dimensions of input features of D and missing data when j is empty. In summary, the learning structure need predict score s and the number of marker places n based on the difference of features D, i.e. $(s, n) = f(D, \Psi)$, where $f()$ represents the regression model and Ψ are the unknown parameters.

Essentially, we choose regression tree as the learning structure due to its imitation for manually judging procedures and its high capability in handling missing and high dimensions features. Besides, regress tree is fast to achieve the robust results, which make the proposed EvaToon platform more

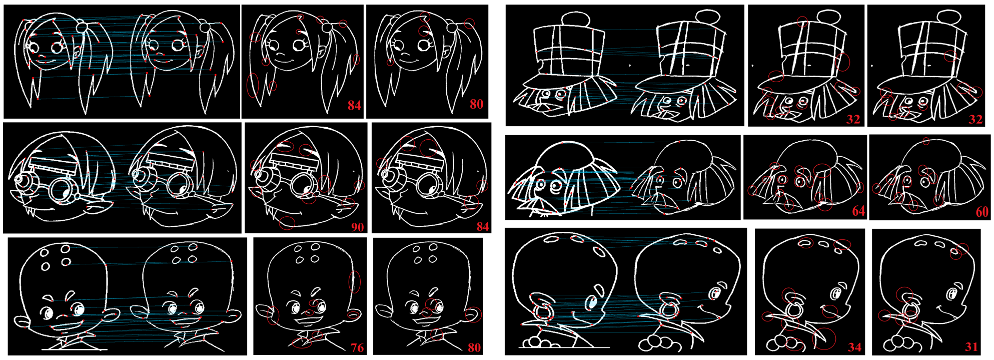

Fig. 5. The matching and prediction results produced by our EvaToon platform. For each result, the first two columns represent the matching results between the original and an imitated drawings, while the third column represents manual evaluations by cartoon experts, including marks and scores on images. The last column shows prediction results of our platform.

efficient. Specifically, we choose CART (Classification And Regression Tree) to be $f()$. Parameters Ψ are determined in non-leaf nodes by making splits through minimizing MSE (Mean Square Error), i.e. $z = \arg\min_{z \in d_i} \sum_{\lambda=1}^{\tau} (y_{z,\lambda} - \tilde{m}_z)^2$, where $d_i = \{d_{i,\tau} | 0 \leq \tau \leq n_e\}$, z is one dimension of d_i, y_z is the feature value of z and $\tilde{m}$ is the mean value, i.e. $\tilde{m}_z = \frac{1}{\tau} \sum_{\lambda=1}^{\tau} y_{z,\lambda}$. For each leaf node $\mathcal{L}$, the labeled variables reaching this node are stored, resulting in a score vector $S_{\mathcal{L}}$ and a mark vector $N_{\mathcal{L}}$.

The construction follows the common CART tree framework [14], which recursively starts from set D and ends when stopping conditions are satisfied, e.g., MSE to split drops down below the product between MSE of the corresponding dimension in D and a predefined threshold, or the number of the examples on a fixed node is relatively small. After construction, we perform cost complexity pruning to avoid the overfit of our tree model validated by k-folders ($k = 10$) cross validation technology.

D. Evaluation of Input Imitated Drawings

In this subsection, we describe how to predict scores and locate marks for a new input cartoon drawing for scoring.

As shown in Fig. 2(d), the evaluation of an input imitated drawing I_ρ consists of the following steps after the construction of regression tree: 1) Preprocessing I_ρ to improve its quality, 2) Locating its interested points, 3) Extracting features around the interested points, and matching the features between I_ρ and the original drawing, and 4) Assigning scores and locating marks based on the prediction of the constructed regression tree. For step 4, we pass the corresponding difference of features d_{I_ρ} through the constructed regression tree. Supposing that I_ρ reach a leaf node $\mathcal{L}$, we thus adopt the mean of the values stored in $\mathcal{L}$ as the predicted score and the number of marks, i.e. $s_\rho = f_m(S_{\mathcal{L}}), n_\rho = f_m(N_{\mathcal{L}})$, where $f_m()$ represents the mean function.

We represent marks as circles to point out the position and scale of unexpected places on the input cartoon drawing. With the adaptively determined number of marks, we select the first n_ρ features in $d_{i,\rho}$ with the largest $L2$ distances to decide marks. In other words, there will be n_ρ marks in I_ρ, which are centered at interested points with the largest distance value among the set of $d_{i,\rho}$. The scale of each mark can be adaptively decided by the largest $L2$ distance among its scale features, i.e. $\tilde{l} = \arg\max_l \bigcup_{l=3}^{\beta} |d_{i,l}|$. Note we assign marks in a more global sense in the last few scales.

IV. Experimental Results

During the cooperation with cartoon experts, we collect 3000 imitated drawing examples for 6 categories of cartoon drawings with three experts' manually scores and marks on them. We preprocess the data of labeled variables by averaging experts' scores to construct the set S, and aggregate different kinds of marks to construct the set Θ, which is described by number, location and the size of marks. Based on these data, we utilize the mean of MSEs after k-folder cross validation to evaluate the accuracy of the prediction of the constructed regression tree:

$$E(f) = \frac{1}{m * k} \sum_{j=1}^{k} \sum_{i=1}^{m} (f(d_{i,j}) - y_{i,j})^2, d_{i,j} \in D_j, y_{i,j} \in O_j \tag{9}$$

where $f()$ refers to the regression tree, k is set to 10, m is the number of validation set, and y represents prediction items of regression tree, i.e., the numbers of marks and scores. We also define precision and recall rate of marks to evaluate the accuracy and completeness of our generated marks, i.e $p(l) = \frac{|l_r|}{|l_t|}$ and $r(l) = \frac{|l_r|}{|l_i|}$, where l_r, l_t and l_i are the set of generated correct marks, generated total marks and ground-truth marks (experts' manual marks).

We show sample matching and predicting examples using our proposed EvaToon platform in Fig. 5. The proposed matching discards different kinds of noises such as multi-lines, the shake of lines, and so on. The matched interested points covers most regions of cartoon drawings, which guarantees the completeness of similarity measuring. From the shown imitation examples, we can intuitively notice that the well and poor imitated drawings are verified by manual scores properly. Our proposed method successfully classifies these drawings

TABLE I
STATISTICS OF EVATOON ON MATCH AND PREDICTION

Category	n_i	n_j	n_{ij}	$\tilde{n}_{ij}$	$P(\%)$	$\tilde{P}(\%)$	$E(s)$	$E(\tilde{s})$	$E(n)$	$E(\tilde{n})$	$p(l)(\%)$	$r(l)(\%)$	$t_m(s)$	$t_p(s)$	$t_{all}(s)$
Boy	70.0	53.3	28.9	35.6	**97.4**	85.5	**17.8**	45.6	7.90	17.8	**78.5**	57.1	8.53	0.240	**8.77**
Hatman	90.0	76.9	29.1	35.5	**95.2**	73.0	**21.1**	57.9	13.2	21.2	**74.6**	52.3	8.42	0.247	**8.67**
MonkSide	90.0	75.2	32.6	37.3	**91.1**	87.2	**17.0**	36.9	14.3	23.3	**70.3**	49.6	8.21	0.236	**8.45**
Girl	90.0	61.5	37.4	41.6	**95.6**	82.2	**16.2**	47.6	6.66	18.7	**76.2**	54.2	9.03	0.253	**9.28**
Oldman	70.0	63.6	27.7	32.8	**90.7**	82.1	**22.1**	52.3	13.3	24.0	**69.0**	50.7	7.93	0.220	**8.15**
Monk	90.0	73.1	25.1	34.2	**95.3**	91.6	**23.8**	46.9	7.40	17.1	**73.9**	53.9	8.30	0.247	**8.55**
Average	83.3	67.3	30.1	36.2	**94.2**	83.6	**19.7**	47.9	10.5	22.9	**73.8**	53.0	8.40	0.241	**8.64**

by assigning the similar scores with those manual scores, and predicts improper regions.

Table. I gives the detailed statistics of our EvaToon platform, measured on a 1.7GHz i5 core2 PC with 6 GB of RAM. We design PS-BP to improve the robustness of describing shape by local patterns of lines and its weight ω is set to 0.35 in the system. To verify the effect of our proposed PS-BP feature, we conduct two kinds of experiments: with and without PS-BP feature to match and predict, represented by no tilde and tilde on signs, respectively. We divide the statistics in Table. I to three sub-parts, which are about the accuracy of matching and prediction, and the efficiency of our platform. For matching, we calculate the mean value of the imitated drawings to represent in the table, where n_i and n_j respectively refer to the number of interested points in the original and an imitated drawing, $n_{i,j}$ refers to the number of matched interested points, and P measures the correctness of the matchings as defined by $P = \frac{n_w}{n_{i,j}}$, where n_w represents the number of correct matchings identified manually. For prediction, we compute precision $p(l)$ and recall rate r_l of marks, and utilize Equ. 9 to compute the MSE value on score s and the number of marks n. For efficiency, we concentrate on the mean computing time for each imitation, where t_m and t_p represent the time of matching and predicting, respectively, while t_{all} is the sum of t_m and t_p.

From Table. I, we notice that small values of $E(s)$ and $E(n)$ are achieved by our platform, which proves the generalization ability and high accuracy of our constructed tree on predicting scores and markers based on cartoon matching. Precision of markers $p(l)$ keeps high, proving the marks generated by our platform are convinced to point out improper drawings, while recall $r(l)$ is a bit low, due to experts' concerns beyond the similarity between the original and the imitated drawing. We verify the improvement in robustness with PS-BP feature by both matching and predicting experiments. For matching, we notice that the PD-HOG-only method generates more matches, but introduces more false ones by comparing $\{n_{ij}, \tilde{n}_{ij}\}$ and $\{P, \tilde{P}\}$, respectively. Too many false matchings will greatly decrease the accuracy in prediction since our platform is based on convinced matching results. For prediction, the reduction of dimensions leads to worse regression results, which is proved by a lower value in $E(\tilde{s})$ and $E(\tilde{n})$. Besides, our platform is fast in both matching and prediction, which are represented by low values in t_m and t_p.

V. CONCLUSION

In this paper, we propose a novel evaluation platform for scoring cartoon drawings. After locating contour pixels with high curvature as our interested points, we extract shape-aware features around the interested points, namely, PD-HOG and PS-BP features. We utilize the two features to match interested points between the original and each imitated cartoon drawing, and further construct a regression tree to predict imitation score with annotated marks. Experimental results on a cartoon dataset illustrate the effectiveness of the proposed evaluation platform. To the best of our knowledge, this is the first work on evaluating cartoon drawings. Our future work includes the exploration on drawing scoring on other types such as Chinese painting and oil painting.

REFERENCES

[1] M. Bessmeltsev, W. Chang, N. Vining, A. Sheffer, and K. Singh, "Modeling character canvases from cartoon drawings," *ACM Trans. Graph.*, vol. 34, no. 5, p. 162, 2015.

[2] A. Bousseau, J. P. O'Shea, F. Durand, R. Ramamoorthi, and M. Agrawala, "Gloss perception in painterly and cartoon rendering," *ACM Trans. Graph.*, vol. 32, no. 2, p. 18, 2013.

[3] Y. Wu, T. Lu, and J. Song, "A real-time mesh animation framework using kinect," in *PCM*, ser. Lecture Notes in Computer Science, vol. 8294. Springer, 2013, pp. 245–256.

[4] H. Li, G. Liu, and K. N. Ngan, "Guided face cartoon synthesis," *IEEE Trans. Multimedia*, vol. 13, no. 6, pp. 1230–1239, 2011.

[5] H. Fu, S. Zhou, L. Liu, and N. J. Mitra, "Animated construction of line drawings," *ACM Trans. Graph.*, vol. 30, no. 6, p. 133, 2011.

[6] E. Iarussi, A. Bousseau, and T. Tsandilas, "The drawing assistant: automated drawing guidance and feedback from photographs," in *UIST*. ACM, 2013, pp. 183–192.

[7] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, 2004.

[8] J. Yu, D. Liu, D. Tao, and H. S. Seah, "Complex object correspondence construction in two-dimensional animation," *IEEE Trans. Image Processing*, vol. 20, no. 11, pp. 3257–3269, 2011.

[9] B. Jin and W. Geng, "Correspondence specification learned from master frames for automatic inbetweening," *Multimedia Tools Appl.*, vol. 74, no. 13, pp. 4873–4889, 2015.

[10] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 44:1–44:10, 2012.

[11] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[12] Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Commun. ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth, 1984.