

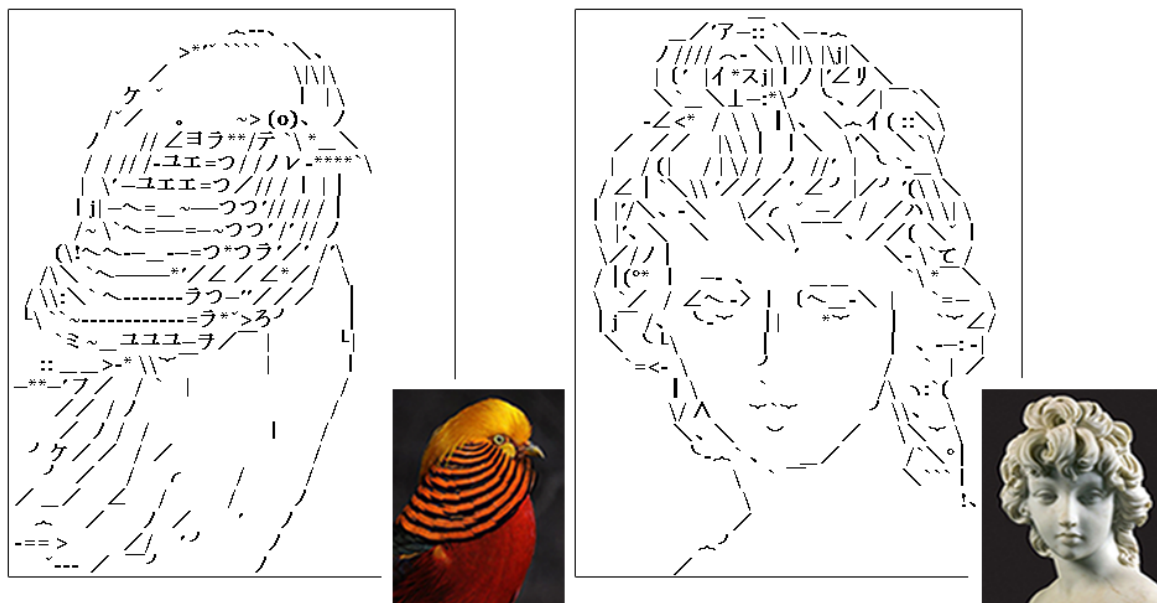
# Texture-Aware ASCII Art Synthesis with Proportional Fonts

Xuemiao Xu<sup>1</sup>, Linyuan Zhong<sup>1</sup> and Minshan Xie<sup>1</sup>, Jing Qin<sup>2</sup>, Yilan Chen<sup>1</sup>, Qiang Jin<sup>1</sup>, Tien-Tsin Wong<sup>3,4,1</sup> and Guoqiang Han<sup>1</sup>

<sup>1</sup>South China University of Technology, China <sup>2</sup>Shenzhen University, China

<sup>3</sup>Computational Perception and Intelligent UI Laboratory of Nanshan District, Shenzhen Research Institute, The Chinese University of Hong Kong

<sup>4</sup>The Chinese University of Hong Kong



**Figure 1:** Our results from real photographs with proportional fonts. Both main contours and textures are well represented.

## Abstract

We present a fast structure-based ASCII art generation method that accepts arbitrary images (real photograph or hand-drawing) as input. Our method supports not only fixed width fonts, but also the visually more pleasant and computationally more challenging proportional fonts, which allows us to represent challenging images with a variety of structures by characters. We take human perception into account and develop a novel feature extraction scheme based on a multi-orientation phase congruency model. Different from most existing contour detection methods, our scheme does not attempt to remove textures as much as possible. Instead, it aims at faithfully capturing visually sensitive features, including both main contours and textural structures, while suppressing visually insensitive features, such as minor texture elements and noise. Together with a deformation-tolerant image similarity metric, we can generate lively and meaningful ASCII art, even when the choices of character shapes and placement are very limited. A dynamic programming based optimization is proposed to simultaneously determine the optimal proportional-font characters for matching and their optimal placement. Experimental results show that our results outperform state-of-the-art methods in term of visual quality.

Categories and Subject Descriptors (according to ACM CCS): I.3.m [Computer Graphics]: Miscellaneous—Visual Art

## 1. Introduction

The wide popularity of modern text-based forums and messaging systems revives the usage of ASCII art. The small text message windows of these systems discourage the adoption of the traditional tone-based ASCII art that generally requires large text resolution for meaningful reproduction. On the other hand, structure-based ASCII art is more favorable as it can reproduce the visual content with a smaller text resolution and more likely to fit in the small text message window. The state-of-the-art method for automatic structure-based ASCII art generation is proposed by Xu et al. [XZW10]. However, their method relies on line drawings as input. This limits its usage for natural images. Moreover, it cannot be extended to proportional font that is commonly used by modern text-based systems.

By observing the artworks from artists (the 4<sup>th</sup> row in Fig. 9), it is found that artists prefer to represent the visually sensitive features in ASCII art results, including both the salient contours and textural structures, since these structures can make the results more appealing. However, even with the state-of-the-art edge detectors, there is no guarantee that visually sensitive edges can always be identified and nonsignificant details can be suppressed. In particular, most modern contour detectors [AMFM11, DZ13] tend to remove most of textures for obtaining the clear contour maps. But, creating ASCII art according to such a contour map may leave many vivid textural structures out, resulting in a dull result that cannot faithfully represent the input image (compare Fig. 2 (a) and our results in the Fig. 1).

In this regard, it is essential to find a feature extraction scheme that is able to reproduce the perceptually prominent structures from the input images. It can extract both salient contours and textural structures generally corresponding to high-scale anisotropic textures [ZSXJ14], while suppressing less-attended details corresponding to low-scale isotropic textures, e.g. noise. The first idea entering our mind is employing the multi-scale Gabor filter, which can capture perceivable features by human. However, Gabor filter is somehow sensitive to the contrast, leading to the loss of features with low contrast (compare the face in Fig. 2 (c) and our result in Fig. 1). On the other hand, psychological evidence [MB88] shows that the human visual system responds strongly to locations in the visual content where the phase information is highly ordered. At these locations, the arrival phases of Fourier components are maximally similar (*phase congruency*). Visual features identified by phase congruency have been demonstrated to be perceptually salient and contain significantly less amount of “false positives” [MB88].

In this paper, we present a novel texture-aware ASCII art generation method with proportional fonts based on phase congruency. Our method can directly take real images, as well as drawings, as input. By computing the phase congruency maps, we can efficiently compare the input image with the text characters. The phase congruency can identify not

only the complete contours even under low contrast, but also the high-scale anisotropic textural structures, which leads to more appealing ASCII art results. More importantly, in order to eliminate the visually insensitive features (fine isotropic textures) which may lead to the over-crowded ASCII art output, we propose a multi-orientation formulation of phase congruency to reflect structures faithfully and hence avoid the confusion of fine isotropic textural structures and salient structures (Fig. 4 (b)).

To suit our application, we design a novel image similarity metric based on our multi-orientation phase congruency model. During the image matching, we propose a novel point-to-area matching strategy to tolerate the local deformation of structure (i.e. misalignment), and the variation of line widths. Experiments demonstrate that with our phase congruency map and matching strategy, our approach outperforms sophisticated local statistics schemes.

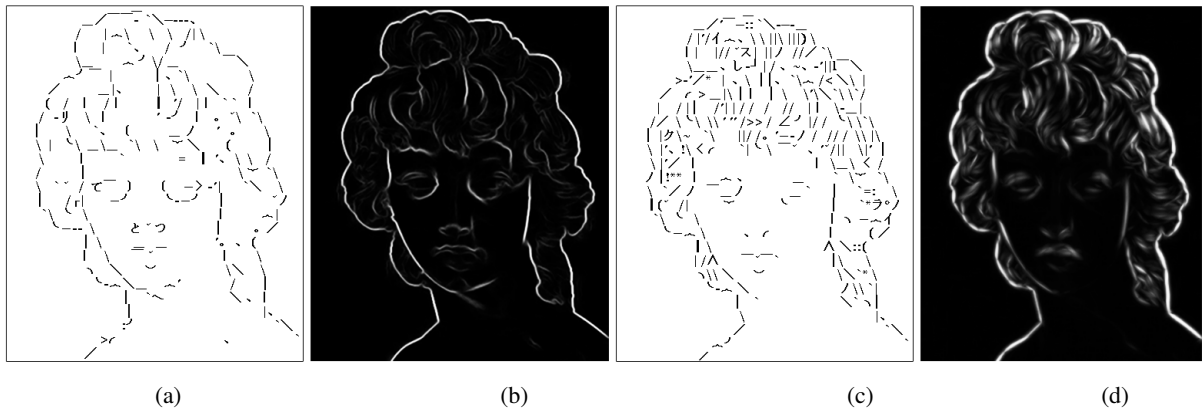
Modern ASCII art generation is further complicated by the popularity of proportional font (different characters can have different width) in most text messaging systems. On one hand, the proportional font complicates our problem as the character placement location is no longer a regular lattice as each row may contain different number of cells (characters). On the other hand, the proportional font offers extra flexibility in placing the characters horizontally (Fig. 7(b) and (d)). Actually, proportional font is crucial for producing ASCII art from natural images, in which complicated textural structures can be well represented by proportional font. In this paper, we simultaneously solve the best character to match and the optimal character placement using dynamic programming.

We demonstrate the effectiveness of the proposed method by taking challenging images, which cannot be faithfully handled by the state-of-the-art methods. User study is also conducted to compare our results and those of other methods. In summary, our contributions include,

- a novel image similarity metric based on our proposed model of phase congruency; the metric can tolerate the misalignment as well as the line width variation; and
- a dynamic programming based optimization that provides the optimal placement of proportional fonts.

## 2. Related Work

**ASCII Art Synthesis** Recent studies on automatic structure-based ASCII art generation can only handle line drawing inputs. Among these works, Martin [Krz11] is the only work that generates proportional-font output. It minimizes the per-pixel difference between characters and the reference image using a greedy approach. Hence, it usually fails to produce pleasant results due to its inability to tolerate deformation and frequently being trapped by the local optimum. Although Miyake et.al [MJN11] proposes to utilize local statistics schemes, such as HOG, to tolerate small local deforma-



**Figure 2:** ASCII art results generated by our framework by replacing the proposed phase congruency model by modern contour detection method [Dollár and Zitnick 2013] (a) and multi-scale Gabor filter (c), (b) and (d) are corresponding feature maps.

tions, its tolerance on large deformation is very limited. To tolerate larger deformations while retaining the overall structure, Xu et al. [XZW10] mimicked the strategies used by ASCII artists, and proposed the alignment-insensitive shape similarity (AISS) metric and a constrained deformation optimization for producing pleasant results. Their work is regarded as the state-of-the-art, and hence we compare our results to theirs in the evaluation.

**Contour/Edge Detector** Early edge detectors, such as Canny and Gabor, care only the local contrast. Hence the edge maps generated by them may introduce false edges or remove positive edges due to the variation of illumination. Sometimes they may also over-strength some visually insensitive features, like minor details and noises, which fail to be represented by characters. Recently, some contour detection methods have been proposed to eliminate textures as much as possible while maintaining main contours in an image. Arbelaez et al. [AMFM11] proposed to extract the key structures and suppress textures by combining local cues in multiple channels into a globalization framework based on spectral clustering. Ren et al. [RB12] and Dollar et al. [DZ13] extended it by exploiting advanced machine learning techniques and produced convincing results. However, it is difficult to synthesize vivid and meaningful ASCII art based on a contour map only containing main contours. In many cases, especially for real photos, perceivably sensitive textural structures are important for the success of ASCII art.

**Image Similarity Metrics** ASCII art generation heavily depends on image similarity measurement. Here, we only survey approaches that account for human perception, including CW-SSIM [SWG\*09], IW-SSIM [WL11], FSIM [ZZMZ11], and HOG [DT05]. For CW-SSIM, IW-SSIM, and FSIM, they are primarily designed for image quality assessment (IQA). The intention is to measure how much deviation from the originals for the images after certain data processing (such as compression). For this specific case, misalignment or character thickness variation are considered as intolerable. However, direct comparison of real

photographs to characters gives rise to large deviation, and our application must tolerate such deviations. As for HOG, its accumulation operation in each bin leads to the high sensitivity to the variation of the character thickness.

AISS [XZW10] is the only metric designed for matching line image to character image. However it only accepts the bitonal input images which are implicitly supposed to contain only salient edges. Moreover, just like HOG, it also suffers the high sensitivity to the character thickness, and limited deformation tolerance. To our best knowledge, no existing metrics can effectively measure the perceptual similarity between real photos and character images.

**Phase Congruency** Phase congruency has been well studied for extracting salient features [MB88, Kov99]. In [Kov99], they proposed a model to extend the theory for phase congruency from the original 1D signal to 2D image. They computed the phase congruency by simply summing the phase congruency values in different orientations. This model has been widely adopted by many image similarity metrics for IQA, such as Liu et al. [LL07] and Zhang et al. [ZZMZ11]. However, this accumulation of local energy along different orientations may easily over-emphasize small texture details and noise, and eventually lead to catching “false positives.” In this paper, we propose a novel model of phase congruency to accurately reflect the image structure, so that it can reliably identify salient structure while suppressing “false positives”.

### 3. System Overview

Fig. 3 overviews our system. It consists of two main steps, feature extraction and optimization for the placement of proportional fonts. To synthesize the ASCII art, our system accepts the reference image and the character data set as inputs. We allow users to specify the target text resolution by defining the desired number of text rows. Optionally, the system can also automatically determine an optimal text resolution. Given the number of text rows and the selected typeface, the

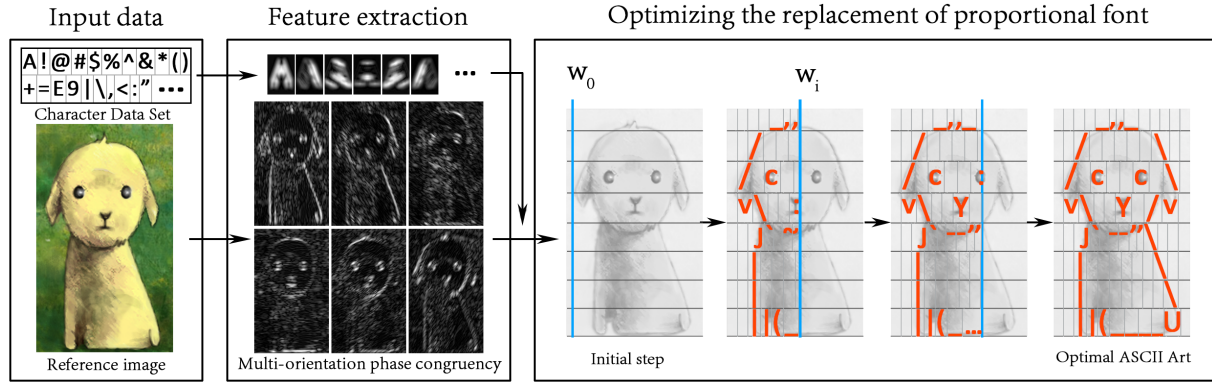


Figure 3: System Overview.

actual height of the output ASCII art can be determined. We then resize the reference image to match the size of the output ASCII image, with the preservation of aspect ratio.

Next, we compute the feature maps for the resized reference image, as well as each character of that typeface, using the proposed phase congruency model (Section 4.1). Note that the phase congruency maps of each particular typeface at particular font size can be precomputed offline and reused for different input images.

Proportional font provides extra flexibility of the character placement which allows us to match the reference image better. However, it complicates the ASCII art synthesis because both the characters to be placed and the placement grid of the characters are unknown. We formulate the determination of the optimal character grid subdivision (placement) and the matching of characters as an optimization to maximize the similarity between the image content and the ASCII image based on the proposed similarity metric (Section 4). To achieve this, we solve our optimization problem using dynamic programming, where we subdivide the placement problem into subproblems recursively, and the final result can be obtained by the combination of the optimal solutions to the subproblems (Section 5). To simplify the computation and achieve high performance, we perform the optimization on each row in parallel.

#### 4. The Similarity Metric

Existing contour detection methods tend to eliminate most of textures in an image, but synthesizing ASCII art based on such a contour map (see Fig. 2 (a) and (b)) ignores the perceptibly sensitive textural structures, resulting unsatisfactory results. Expressing the presence of textures through appropriate characters will make the results more lively and meaningful (see Fig. 1 and more results as well as a user study in Section 6). To the end, we propose a novel multi-orientation phase congruency model to extract both salient contours and textural structures for character matching.

#### 4.1. Multi-Orientation Phase Congruency

The basic idea of phase congruency is that the human visual system responds strongly to the locations in an image where the phase is highly ordered. In fact, such phase congruency can be formulated in different ways. Different applications may favor different formulations.

The original phase congruency model [MB88] is designed for 1D signal  $I(t)$ . It can be efficiently computed based on their band-pass versions  $[e_n(t), o_n(t)] = [I(t) * f_e^n(t), I(t) * f_o^n(t)]$ , where operator  $*$  means convolution, and  $n$  corresponds to scale. For the pair of functions  $f_e(t)$  and  $f_o(t)$ , we adopt the well-known quadrature filter, multi-scale Gabor filter, which gives the optimal compromise between spatial and frequency indetermination [Dau85]. The phase congruency was defined as:

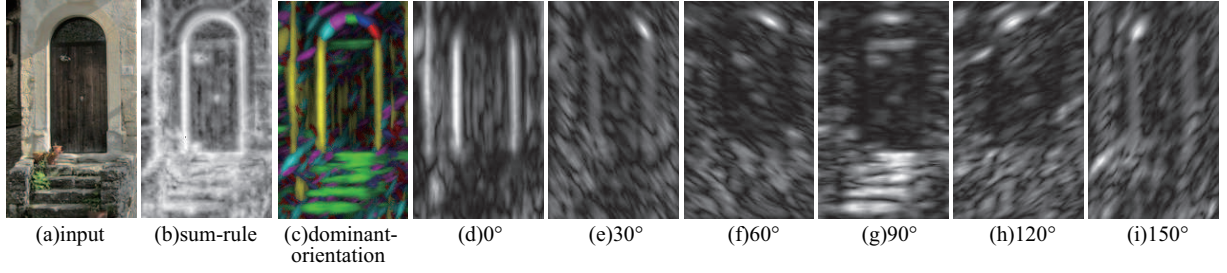
$$P(t) = \frac{E(t)}{\epsilon + \sum_n A_n(t)} \quad (1)$$

where  $E(t) = \sqrt{(\sum_n e_n(t))^2 + (\sum_n o_n(t))^2}$  indicates the local energy;  $A_n(t) = \sqrt{e_n(t)^2 + o_n(t)^2}$  is the amplitude on scale  $n$ ;  $\epsilon$  is a small positive constant to avoid the problem of incorrectly highlighting some unnoticeable visual points when  $\sum_n A_n(t)$  is very small in smooth regions. Note that this original model performs the statistics in a relatively large area, and hence can effectively counteract the effects of local deformations.

To extend the original 1D model to 2D, a popular integration rule is simply summing the energies and amplitudes on all orientations [Kov99]. This summation form has been validated to be effective for salient visual features.

However, this summation scheme tends to over-emphasize some visually insensitive structures, generally corresponding to the low-scale isotropic textures. Such kind of texture often exists in natural images, e.g. rough pavement, meadow, tree bark, and hair. These textures are often highlighted by phase congruency with the summation rule. For example, as shown in Fig. 4 (b), it almost fails to recognize the stair structure as the fine textures on surfaces are





**Figure 4:** Comparison between summation and our multi-orientation formulation. (b) is the summation-based phase congruency map. (c) is the combined phase congruency map with phase congruency value selected from the dominant orientation in (d)-(i). (d)-(i) are our phase congruency maps on six orientations.

over-emphasized. Unfortunately, such textures are typically not important in ASCII art. The fundamental reason of this undesirable over-emphasis is that the accumulation of small local energies on different orientations. In these regions, although the energies along different orientations are not large (Fig. 4 (d)-(i)), the summation of them may be large enough to be highlighted in the phase congruency map (Fig. 4 (b)). In other words, summation reduces the data dimensionality from multiple orientation information to a scalar one, and inevitably leads to certain loss of information.

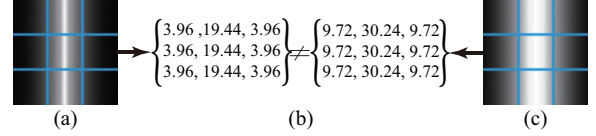
To tackle this problem, we propose a novel approach. Instead of combining the information from different orientations, we propose to keep the phase congruency values of different orientations to form a vector:

$$F(x, y) = \left\{ P_{\theta_j}(x, y), j \in [0, 5] \right\} \quad (2)$$

$$P_{\theta_j}(x, y) = \frac{E_{\theta_j}(x, y)}{\epsilon + \sum_n \sum_j A_{n, \theta_j}(x, y)},$$

where  $P_{\theta_j}(x, y)$  is the phase congruency value on orientation  $\theta_j$ . Based on the study of phase congruency, the phase congruency value of an orientation can faithfully reflect the perceptual-correlated strength on this particular orientation. By retaining the value from each orientation, we can better retain the perception information. The question is how many orientations we should retain. According to neurophysiological findings, human vision is sensitive to specific orientations with approximate bandwidths of  $\pi/6$  [Lee96]. In particular, we retain six orientations which are evenly distributed over  $\pi$ , i.e.  $\theta_j = j\pi/6$ , where  $j \in [0, 5]$ . Fig. 4 (d)-(i) show the phase congruency maps of the six orientations.

Fig. 4 (c) visually demonstrates the advantages of keeping phase congruency values of different orientations compared to the summation rule. It is clear that the major structure in the map corresponding to the dominant orientation is more apparent. As illustrated in our multi-orientational maps (Fig. 4 (d)-(i)), the vertical lines of door are highlighted in the map of orientation  $0^\circ$ , and the horizontal lines of steps are highlighted in the map of orientation  $90^\circ$ . In contrast, the pixels in fine texture regions have small values in all orientations. Thus, the phase congruency vector for each pixel can faithfully reflect a variety of structures, and avoid the confusion of fine textural structures and salient structures.



**Figure 5:** (a) and (c) are the phase congruency maps (orientation  $0^\circ$ ) computed from the input images of vertical lines with thickness of 1 and 4 pixels, respectively. Suppose the maps are divided into  $3 \times 3$  cells. Their histograms in (b) could be very different.

## 4.2. Deformation-Tolerant Matching

As the input image can be arbitrary while the choices of character shapes and placement are very limited, it is impossible to always match the input with exactly aligned characters. Besides, another type of deformation is introduced by the typeface, as the same character in different typefaces may be different in thickness and style. Thus, deformation tolerance is a must during the matching.

To tolerate the deformation, one popular way is to apply the local statistics schemes, such as the block diagram in HOG [DT05], log-polar diagram in SC [BMP02] and AISS [XZW10]. However, the drawback is their high sensitivity to the character thickness or style. As shown in Fig. 5, the statistical values change dramatically with the change of line thickness. Hence, the local statistics schemes may not be suitable for ASCII art synthesis from arbitrary images and character fonts.

To tolerate the style and line thickness variations of different typefaces, we propose to perform the matching in a point-to-area (PTA) mode which finds the best matched point in a small neighborhood. This nature of area expansion allows structure to be deformed in a small scale. Since PTA only concerns the best matched point in a small neighborhood instead of accumulating the values in the same neighborhood, it is robust to the variation of line thickness. With this matching strategy, every point in one map in Fig. 5, can find a well matched point within a neighborhood with radius  $r$  on the other map. Its matched point can be calculated by:

$$F'(x_m, y_m) = \arg \min_{s, t} \{ D_P(F(x, y), F'(x + s, y + t)) \} \quad (3)$$

where  $F$  and  $F'$  are the two feature maps. For a give point  $(x, y)$  in  $F$ ,  $(x_m, y_m)$  is the matched point in the feature map  $F'$ . The  $s, t$  belong to  $(-r, r)$ , and we set  $r$  as  $1/6$  of the characters' height, which is 3 pixels in our experiments.  $D_P(F(x, y), F'(x, y))$  is the distance between the two six-component phase congruency feature vectors (Eq.2).

The distance between the two feature vectors can be calculated as the distance between two histograms. However, the difference between orientation may be lost by directly use bin-to-bin comparison scheme like Euclidean distance. Instead, we adopt the cross-bin scheme, Earth Mover Distance [PW08] for the calculation, in which both of the orientation and phase congruency value can be taken into account.

Finally, the dissimilarity  $D_{DSM}$  between two feature maps  $S$  and  $S'$  is further measured by calculating the distances from  $S$  to  $S'$  and from  $S'$  to  $S$ , respectively. This bidirectional measurement is more stable. The formulation is given by:

$$\frac{1}{2} \sum_{x=1}^W \sum_{y=1}^H (D_P(F(x, y), F'(x_m, y_m)) + D_P(F'(x, y), F(x_m, y_m))). \quad (4)$$

where  $W$  and  $H$  are the width and height of the feature maps.

### 4.3. Comparison with Existing Metrics

To validate our metric, we compare it to several related state-of-the-art image similarity metrics, including HOG, FSIM, and AISS. HOG is a popular metric which aims at detecting the "human" from natural images. FSIM is the recently proposed perception-based metric for image quality assessment, which is also based on phase congruency. AISS is the only similarity metric dedicated for ASCII art application. Note that the original AISS is only applicable for comparing iso-thickness line drawings, we compute AISS based on the local maxima map generated by the original phase congruency model, and call it "AISS+TPC\_Map".

Fig. 6 compares the results of our metric, HOG, FSIM and AISS. In the first query image, character " $\wedge$ " is slightly deformed in structure. HOG, AISS and our metric can successfully find the correct character since all of them can tolerate deformation, while FSIM fails since it assumes the compared images are well-aligned. In the second query, the query image contains strong noise in background. HOG, FISM and AISS all fail to extract the major structure " $V$ " from the background. In contrast, our metric can successfully recognize the character " $V$ " since it can reflect the structure faithfully and hence clearly distinguish the noise and salient structures. In the third query, since the query image is a thickened vertical line " $|$ ", HOG and FISM cannot find the correct character due to their high sensitivity to the line thickness. On the other hand, our method is robust to the line thickness. AISS can also find the correct character because its local maxima map of phase congruency only consists of centerlines. In the fourth query, the query image contains a  $45^\circ$  slash. Our metric can successfully find the





















Query	Our metric	HOG	FSIM	AISS +TPC_Map
				
				
				
				

Figure 6: Comparison with existing metrics

more reasonable character " $/$ ," since our metric can measure the deformation accurately. However, other metrics find the character " $<$ ." That is because the  $45^\circ$  slash and character " $/$ " fall into two different bins while the  $45^\circ$  slash and " $<$ " partially fall into the same bin in these metrics.

### 5. Optimal Matching with Proportional Font

When reproducing the reference image with the proportional fonts, we need to simultaneously solve two subproblems; they are the choices of characters and their locations. This is a complex problem as the choice of one character affects both the choices, as well as, the position of the subsequent characters as we place characters from left to right.

We solve this 2D-ASCII art matching problem by subdividing it into  $N$  1D matching problems, where  $N$  is the number of text rows. This approach simplifies our problem by turning a 2D optimization into  $N$  independent 1D optimization problems, and at the same time, allows parallelization.

Even for this 1D optimization, naive greedy search is not satisfactory. Fig. 7 (c) shows the result using greedy search. In this greedy search, we find the best-matched characters to represent each row of the reference image from left to right sequentially. But it is easily trapped into the local optimum, leading to the result of the whole image is not optimal. Instead, we solve this 1D optimization by dynamic programming. We subdivide the problem into a series of subproblems recursively as illustrated by Fig. 8. In Fig. 8 (top), we divide the problem of matching the reference region into two subproblems, i.e. finding an optimal character  $c_i$  (single character) to represent the rightmost part of the region, and finding an optimal solution (a sequence of characters) to represent the left part of the region. Similarly, we can recursively subdivide the later subproblem into even smaller subproblems (Fig. 8 (middle)). The recursive problem subdivision continues until the leftmost part has the same width  $w_0$  of the thinnest character in our character library (Fig. 8 (bottom)).

Suppose that an array element  $A[w]$  is the optimal sequence of characters to represent the reference region with the width  $w$ . It corresponds to an array element  $M[w]$  that



**Figure 7:** (a) the reference image, (b) Fixed-width font, (c) proportional font with greedy approach, (d) proportional font with dynamic programming

**Algorithm 1** The proposed optimization algorithm based on dynamic programming.

**Input:** the reference image  $I$ , character image set  $C[1..K]$   
**Horizontal Partitioning:** partition  $I$  into  $N$  Stripes  $I_1 \dots I_N$

```

parfor  $i=1:N$  do
  Initialization:  $M_i[1..W] = \infty, A_i[1..W] = \{\}$ 
  for  $\omega = 1:W$  do
    for  $k=1:K$  do
       $\omega' = \omega - C[k].width$ 
      if  $M_i[\omega] > D_{DSM}(I_i[\omega], R(A_i[\omega'] \oplus C[k]))$  then
         $M_i[\omega] = D_{DSM}(I_i[\omega], R(A_i[\omega'] \oplus C[k])), A_i[\omega] =$ 
         $A_i[\omega'] \oplus C[k]$ 
      end if
    end if
  end for
end for
end parfor

```

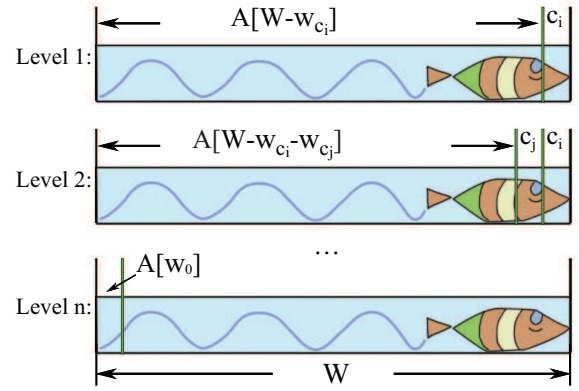
represents the optimal cost of using the sequence of characters  $A[w]$  to represent the reference region with the width  $w$ , where  $w \in [0, W]$  and  $W$  is the width of the reference image. Our algorithm calculates  $A[w]$  in each iteration and ultimately obtains the optimal solution for the whole reference region. We calculate  $M[w]$  in each iteration by,

$$M[w] = \begin{cases} \min_{i \in [1, K]} \{E(A[w - w_{c_i}], c_i)\} & s.t., w - w_{c_i} \geq 0 \\ 0 & otherwise. \end{cases} \quad (5)$$

where  $c_i$  is the  $i^{th}$  character and  $i \in [1, K]$ , where  $K$  is the total number of distinct characters available;  $w_{c_i}$  is the width of the character  $c_i$ . Objective function  $E(A[w'], c_i)$  computes the matching cost using the character sequence by appending  $A[w']$  with  $c_i$  and defined as,

$$E(A[w'], c_i) = D_{SM}(I(w), R(A[w'] \oplus c_i)) \quad (6)$$

where  $D_{SM}$  is image similarity defined by Eq. (4), operator  $\oplus$  concatenates two character sequences,  $I(w)$  is the current



**Figure 8:** Optimal substructure: Our problem can be broken into simpler subproblem recursively.

reference region of width  $w$ ; and  $R(c)$  is the rasterized image of the character sequence  $c$ .

We start the optimization by swiping the reference image from left to right as illustrated in the right half of Fig. 3. It starts from the smallest problem, i.e. a reference region with the same width as the width of the thinnest character  $w_0$ . Then we advance the problem by increasing the width by 1 pixel until it reaches to the right boundary of the reference image, i.e. width  $W$ . Inside each iteration, we calculate a new  $A[w]$  based on the new input and current  $A[w]$  using our algorithm. Algorithm 1 gives the pseudocode of the proposed optimization algorithm. Fig. 3 illustrates the optimization process. Fig. 7 visually compares our results to that of the fixed width ASCII art, and the proportional font ASCII art with greedy strategy.

## 6. Results and Discussion

To validate our method, we conducted multiple experiments over a rich variety of inputs including hand-drawn images and natural images (Fig. 1, Fig. 7, Fig. 9 and Fig. 10; more results are provided in the supplementary materials). Our

method accepts any typefaces. As our method allows for parallelization, it is quite fast and can synthesize a large ASCII art image with text resolution  $50 \times 50$  within one minute.

**Comparison to existing methods** To the best of our knowledge, our work is the first attempt to generate the structure-based ASCII art from arbitrary images with proportional font. First, to compare the visual quality of ASCII art results, we apply our framework based on the contour map generated by a state-of-the-art contour detector [DZ13]. Moreover, the state-of-the-art structure-based ASCII art generation method proposed by Xu et al. [XZW10] relies on line drawings as input. For a fair comparison, the results by Xu et al. [XZW10] are generated by feeding their method with our structure map as input. In addition, we also compare to the ASCII art manually created by artists. We label these three competitors as “Cont.,” “Xu” and “Artist.”

Fig. 9 shows the results of the four methods, and their input images have been included in Fig. 10. Compared to “Cont.” results shown in the 2<sup>nd</sup> row, it is observed that the results of our method are more visually appealing, as more visually sensitive texture patterns are well preserved and vividly represented by the proportional font. Check the textures in the scarf of the girl, the house and fence, and particularly the feather of the lion. In contrast, the result of “Cont.” on the lion image is too blank and dull to represent the mien of the lion. When compared to the results of “Xu” shown in the 3<sup>rd</sup> row, our results are much more identifiable and pretty. This is because our method fully exploits the flexibility of the proportional font, which can better represent the variety of texture patterns, while Xu et al. [XZW10]’s method can only support fixed width fonts. Of course, the creations of artists (4<sup>th</sup> row) are the most stylish and interesting as they can intelligently deform and even modify the content for reproduction. Fig. 10 shows more results of our method from natural images and drawings with variety of textures. It is observed that our method can preserve both contours and textural structures in these images, and well represent them by proportional font. Most results are clear, meaningful and vivid.

**User study** We further conducted a user study to evaluate our results and those of our competitors. We prepared 29 test sets covering a wide range of images: 6 bitonal line drawings, 10 cartoon images, and 13 real photographs. In most cases, three automatic methods are compared, including our method, “Cont.” and “Xu.” Among them, 5 cases also includes the manual works by artists. Note that the order of the results by the four methods is randomly organized, and the reason that we only manage to have 5 artworks is because the manual creation of ASCII art is very time-consuming and difficult. Users are asked to grade the results in the range of [1,10] in terms of similarity, recognition ability and aesthetics. This user study has been posted on website and available to online participants. All the test sets can be found in

	Line images	Cartoon images	Photos
Ours	8.65/8.48/8.38	8.41/8.23/8.31	8.72/8.61/8.46
Cont.	8.18/7.48/8.16	7.28/7.64/6.62	6.26/6.12/5.86
Xu	6.91/6.76/6.60	5.50/5.23/5.19	5.63/5.36/5.23
Artist	8.04/8.1/8.51	8.21/8.20/8.30	8.63/8.48/8.35

**Table 1:** Average scores of the user study:  $n_1/n_2/n_3$  indicate the average scores of similarity, recognition ability and aesthetics respectively.

	Line images	Cartoon images	Photos
Ours	1.10/1.08/1.17	1.52/1.09/1.09	1.48/1.44/1.09
Cont.	2.09/1.62/0.98	1.62/0.82/1.00	0.98/1.02/1.36
Xu	0.86/0.83/1.24	1.27/0.75/0.67	1.40/0.71/1.30
Artist	2.13/1.31/1.51	1.85/1.47/0.74	1.52/1.01/1.23

**Table 2:** The standard deviations of the user study:  $n_1/n_2/n_3$  indicate the standard deviation of similarity, recognition ability and aesthetics respectively.

the supplementary materials. There are 65 participants who have taken part this evaluation.

Table 1 and Table 2 show the average scores and the corresponding standard deviations of different methods. Table 3 shows the analysis results on the difference of the scores between our method and the competitors using t-test. It is observed in Table 1 that, for line drawings, the results of three automatic methods are all acceptable (all scores are more than 6.0). Nevertheless, our method achieves the highest scores and outperforms the other two methods. For cartoon images and real photos, our results get significant higher scores than other two. The scores of the “Cont.” drop because the contour detection method attempts to remove the textures in the input images as much as possible, resulting in dull ASCII art results. The score dropping of “Xu” demonstrates that the fixed-width font cannot well represent a variety of structures in natural images. Compared to manual artworks, our method is less creative than humans. Artists can intelligently modify or even drop the content in order to facilitate the ASCII art composition. It is also observed in Table 3 that in most cases there is a significant difference of the scores between our method and the competitors ( $p < 0.05$  for the t-test). All these evidence the effectiveness of our method in the similarity measurement and optimal matching quality of proportional font.

**Animated ASCII art** We further extend our method to generate the ASCII art animation. To guarantee the consistency of character images in continuous frames, we try to separate the background and front regions, and then convert them to ASCII art images individually using our method, and finally merge them together. In this way, the background of ASCII art are consistent during the animation. One animated result has been included in the supplementary materials.

**Limitation** Our work has an obvious limitation that it considers no temporal coherence of front regions when it is applied to ASCII art animation. The second limitation is that when the specified text resolution is too low to repre-





**Figure 9:** Comparison of the results by four methods. From top to down: the results generated by our method, "Cont.", "Xu" and "Artist".



Figure 10: Results

	Line images	Cartoon images	Photos
Cont.	no/no/yes	yes/yes/yes	yes/yes/yes
Xu	yes/yes/yes	yes/yes/yes	yes/yes/yes
Artist	no/yes/yes	yes/no/yes	yes/yes/no

**Table 3:** Statistic analysis using *t*-test:  $n_1/n_2/n_3$  indicate whether  $p < 0.05$  when comparing our scores with other methods' scores in terms of similarity, recognition ability and aesthetics. The  $p < 0.05$  means a significant difference between the scores.

sent some texture structures in the given image, the clarity of the resultant ASCII art may be hurt. In addition, we do

not take into account of color, although color text is available on many text-based systems.

## 7. Conclusion

We propose a novel structure-based ASCII art generation method that accepts arbitrary images, and it is the only automatic method that can generate proportional-font ASCII art. To achieve these, we propose a novel image similarity metric based on multi-orientation phase congruency model, which is proposed to maintain visually salient structures, including both main contours and textural structures, while suppress-

ing visually unimportant details. Compared to existing similarity metrics, our metric can tolerate misalignments as well as character thickness and style. A dynamic programming based optimization is used to simultaneously solve the character matching and placement problems. Extensive experiments on a rich variety of inputs and user study demonstrate the effectiveness of our method. Our similarity metric may naturally be extended to other NPR applications.

## Acknowledgments

This work was supported by grants of NSFC and NSFC-Guangdong (Grant No. 61103120, 61472145, 61272293, 61233012 and S2013010014973), Doctoral Program of Higher Education of China (Grant No. 20110172120026), Shenzhen Nanshan IIEF(Grant No. KC2013ZDZJ0007A), Shenzhen Basic Research Project (Grant No. JCYJ20120619152326448), RGC of Hong Kong (Grant No. 417913), Guangzhou Novo Program of Science and Technology (Grant No. 0501-330) and Guangzhou Key Lab of cloud computing technology and safety evaluation.

## References

- [AMFM11] ARBELAEZ P., MAIRE M., FOWLKES C., MALIK J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 5 (2011), 898–916. URL: <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.161>, doi:10.1109/TPAMI.2010.161.2,3
- [BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 4 (2002), 509–522. URL: <http://doi.ieeecomputersociety.org/10.1109/34.993558>, doi:10.1109/34.993558.5
- [Dau85] DAUGMAN J. G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Optical Society of America, Journal, A: Optics and Image Science* 2, 7 (1985), 1160–1169. doi:10.1364/JOSAA.2.001160.4
- [DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), pp. 886–893. URL: <http://dx.doi.org/10.1109/CVPR.2005.177>, doi:10.1109/CVPR.2005.177.3,5
- [DZ13] DOLLÁR P., ZITNICK C. L.: Structured forests for fast edge detection. In *IEEE International Conference on Computer Vision* (2013), pp. 1841–1848. URL: <http://dx.doi.org/10.1109/ICCV.2013.231>, doi:10.1109/ICCV.2013.231.2,3,8
- [Kov99] KOVESI P.: Image features from phase congruency. *VIDERE: Journal of computer vision research* 1, 3 (1999), 1–26. doi:10.1.1.4.1641.3,4
- [Krz11] KRZYWINSKI M.: Ascii art proportional spacing, tone/ structure mapping and fixed strings, 2011. URL: <http://mkweb.bcgsc.ca/asciiart/>.2
- [Lee96] LEE T. S.: Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 10 (1996), 959–971. URL: <http://doi.ieeecomputersociety.org/10.1109/34.541406>, doi:10.1109/34.541406.5
- [LL07] LIU Z., LAGANIÈRE R.: Phase congruence measurement for image similarity assessment. *Pattern Recognition Letters* 28, 1 (2007), 166–172. URL: <http://dx.doi.org/10.1016/j.patrec.2006.06.019>, doi:10.1016/j.patrec.2006.06.019.3
- [MB88] MORRONE M. C., BURR D. C.: Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society of London.* (1988), 221–245. doi:10.1098/rspb.1988.0073.2,3,4
- [MJN11] MIYAKE K., JOHAN H., NISHITA T.: An interactive system for structure-based ascii art creation. In *Proc. of NICOGRAPH International 2011* (June 2011).2
- [PW08] PELE O., WERMAN M.: A linear time histogram metric for improved SIFT matching. In *Proc. Computer Vision - ECCV*. 2008, pp. 495–508. URL: [http://dx.doi.org/10.1007/978-3-540-88690-7\\_37](http://dx.doi.org/10.1007/978-3-540-88690-7_37), doi:10.1007/978-3-540-88690-7\_37.6
- [RB12] REN X., BO L.: Discriminatively trained sparse code gradients for contour detection. In *26th Annual Conference on Neural Information Processing Systems*. (2012), pp. 593–601.3
- [SWG\*09] SAMPAT M. P., WANG Z., GUPTA S., BOVIK A. C., MARKEY M. K.: Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing* 18, 11 (2009), 2385–2401. URL: <http://dx.doi.org/10.1109/TIP.2009.2025923>, doi:10.1109/TIP.2009.2025923.3
- [WL11] WANG Z., LI Q.: Information content weighting for perceptual image quality assessment. *IEEE Transactions on Image Processing* 20, 5 (2011), 1185–1198. URL: <http://dx.doi.org/10.1109/TIP.2010.2092435>, doi:10.1109/TIP.2010.2092435.3
- [XZW10] XU X., ZHANG L., WONG T.: Structure-based ASCII art. *ACM Trans. Graph.* 29, 4 (2010). URL: <http://doi.acm.org/10.1145/1833351.1778789>, doi:10.1145/1833351.1778789.2,3,5,8
- [ZSXJ14] ZHANG Q., SHEN X., XU L., JIA J.: Rolling guidance filter. In *Proc. of Computer Vision - ECCV* (2014), pp. 815–830. URL: [http://dx.doi.org/10.1007/978-3-319-10578-9\\_53](http://dx.doi.org/10.1007/978-3-319-10578-9_53), doi:10.1007/978-3-319-10578-9\_53.2
- [ZZMZ11] ZHANG L., ZHANG L., MOU X., ZHANG D.: FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing* 20, 8 (2011), 2378–2386. URL: <http://dx.doi.org/10.1109/TIP.2011.2109730>, doi:10.1109/TIP.2011.2109730.3