Layout Style Modeling for Automating Banner Design

Yunke Zhang*
Hangzhou Normal University
Alibaba Group
el.luke.chang@gmail.com

Kangkang Hu
Alibaba Group
kangkang.hkk@alibaba-inc.com

Peiran Ren
Alibaba Group
peiran.rpr@alibaba-inc.com

Changyuan Yang
Alibaba Group
changyuan.yangcy@alibaba-inc.com

Weiwei Xu
Zhejiang University
xww@cad.zju.edu.cn

Xian-Sheng Hua
Alibaba Group
xiansheng.hxs@alibaba-inc.com

ABSTRACT

Banner design for is challenging to clearly convey information while also satisfying aesthetic goals and complying with the banner owner or advertiser's visual identity system. In online advertising, banners are often born with tens of different display sizes and rapidly changing design styles to chase fashion in many distinct market areas and designers have to make huge efforts to adjust their designs for each display size and target style. Therefore, automating multi-size and multi-style banner design can greatly release designers' creativity. Different from previous work relying on a single unified omnipotent optimization to accomplish such a complex problem, we tackle it with a combination of layout style learning, interpolation and transfer. We optimize banner layout given the style parameter learned from a set of training banners for a particular display size and layout style. Such kind of optimization is faster and much more controllable than optimizing for all sizes and diverse styles. To achieve multi-size banner design, we collect style parameters for a small collection of various sizes and interpolate them to support arbitrary target size. To reduce the difficulty of style parameter training, we invent a novel style transfer technique so that creating a multi-size style becomes as easy as designing a single banner. With all of the three techniques described above, a robust and easy-to-use layout style model is built, upon which we automate the banner design. We test our method on a data set containing thousands of real banners for online advertising and evaluate our generated banners in various sizes and styles by comparing them with professional designs.

CCS CONCEPTS

• **Computing methodologies → Shape modeling**; Rendering; Image manipulation; • **Theory of computation** → *Computational geometry*;

*This work was done when the corresponding author was visiting Alibaba as a research intern.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ThematicWorkshops'17, October 23–27, 2017, Mountain View, CA, USA.
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5416-5/17/10…$15.00
https://doi.org/10.1145/3126686.3126718

KEYWORDS

Banner Layout, Multi-size Style, Layout Optimization, Style Interpolation, Style Transfer

1 INTRODUCTION

Banner design for advertisement is ubiquitous in modern life. Creating and refining designs can be time consuming and involves professional skills to convey information clearly while also satisfying aesthetic goals. Although creating an original design is more artistic, we argue that creating a series of them with consistent visual appearance must comply with some measurable rules. The latter is very common when designing for a wide variety of display sizes or creating a bunch of designs in various styles for different products advertised either online or offline. We call this multi-size and multi-style design problem as series design. Traditionally, designers have to manually generate and modify banners for each size and style and their efforts on supporting multi-size and multi-style can be 10 to 100 times more than designing a new banner. As the need of new design keeps growing significiantly, automating series design not only significantly reduces the effort of designers, but also encourages new ideas and educates novice users [5].

In this paper, we consider series banner design for advertisements of various products. Banners usually consist of a small number of elements, either text or image, and these elements often share some common roles, *e.g.* title, product, logo, tag and background. While there is previous research on automatically creating single page graphic design [19], there is little work on series design emphasizing multi-size and multi-style support. Though banner design is a complex problem involving layout, colors, fonts and more, we specifically focus on the layout aspect and leave others for future work, since layout mutates much more than others when banner size changes, which also makes it the most crucial part of series banner design.

The layout problem is defined as specifying the locations and sizes of all elements in a banner. And we assume that a set of texts and images are provided as inputs along with associated meta-data, *e.g.* their roles. Our target is to arrange these elements properly, in a user specified style.

Instead of inventing an omnipotent energy function [19] and optimizing it to design layout in arbitrary size and style, we develop a more controllable scheme with a combination of three techniques as the fixed-size layout style learning, style interpolation among

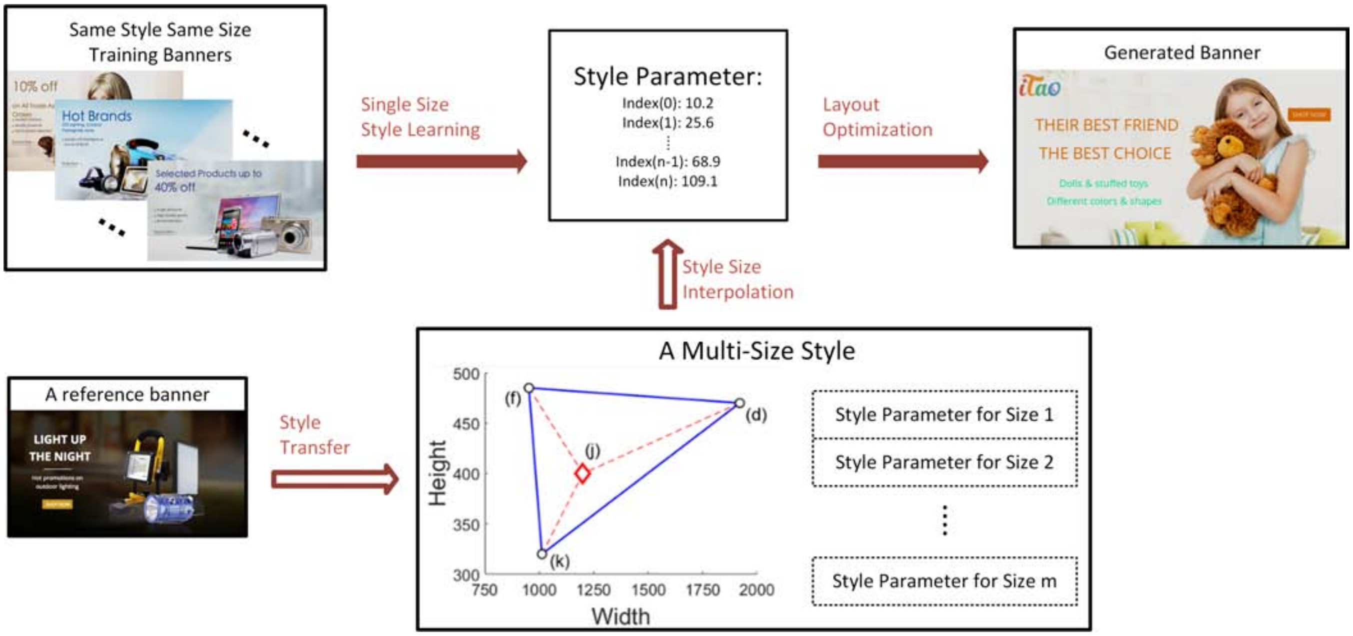

Figure 1: System pipeline. The basic pipeline is learning a single size style from a batch of training banners, then the learned style parameter can be used to optimize new banners. To support multi-size banner design, we manage a multi-size layout style with a small collection of style parameters of different banner sizes and interpolate it to generate the target size style parameter. This multi-size layout style can be modified to mimic significantly different designs with our style transfer technique given a few or even a single reference banners.

different sizes and (multi-size) style transfer with minimal reference banner exemplars. We name our approach as the layout style modeling and show our system pipeline in Figure 1.

Design style is a subjective concept and there is hardly any rigid boundary between two "adjacent" style instances. The situation can be even worse when dealing with multiple display sizes. Our methodology is to handle each size separately by learning a probabilistic model on a small set of aesthetic quantities. Once learned, one can design a banner by optimizing an energy function derived from the probabilistic model. Style parameter (a vector) used in this optimization characterizes the feature of that layout style in a specific banner size, and we model a multi-size style using a collection of style parameters in various sizes. To acquire the style parameter of a banner size outside a collection, we interpolate neighboring banner sizes. However, interpolation is based on smoothness hypothesis, which may not be guaranteed all the time. In this paper, discontinuity detection is applied upon which we develop our interpolation technique.

Training a statistically sound multi-size style requires hundreds to thousands of exemplars as training data. It would be much better if we can infer a new style by modifying an existing one with only one or a very small number of new banner exemplars. For this purpose, we develop a simple but safe scheme to proliferate new styles. We believe this technique can greatly assist designers to create new ideas and process large amount of designs in their daily work.

To our knowledge, this paper is the first that considers banner layout as a multi-size style modeling problem. Our key technical contributions include:

- A simple probabilistic model for layout style using Gaussians defined on a minimal set of aesthetic quantities. With this model, one can create a banner layout by optimizing an energy-based function in less than a second.
- A practical style interpolation approach considering mutation (discontinuity) among different display sizes inside a multi-size layout style.
- An efficient layout style transfer method to reduce difficulties for generating diverse multi-size styles with minimal banner exemplars.

We test our method on real banner designs, producing competitive results vs those created by professional designers. Our banner layout system will be released in large e-commerce banner design system soon.

In the rest of this paper, we will first investigate prior work in Section 2 and clarify our problem definition in Section 3, then describe how to learn a fixed-size style in Section 4, and discuss our style interpolation and transfer techniques in Section 5 and 6 respectively. Finally, we show results in Section 7 and conclude in Section 8.

2 RELATED WORK

Previous researchers have worked on automating layout for years. However, there is little work targeting automatic banner layout.

The most related problem is the single-page layout problem [19–21], involving placing text and image elements without a document structure, *e.g.* Document Object Model (DOM). Purvis *et al.* [21] use genetic algorithm to optimize an energy function considering alignment and balance when placing text and figure blocks. O'Donovan *et al.* [19] present an energy-based model for single-page layout and accelerate their method for interactive design [20]. Their cost function has a similar formulation to ours, however their formulation is much more complex while our formulation is derived from a probabilistic model and justified with real data analysis. Moreover, it is unclear how well they can apply their method for layouts in various sizes and styles. Although they claim to learn model parameter from exemplars by nonlinear inverse optimization (NIO) [14], the initial parameter is manually set which may seriously impact their nonlinear optimization results. Instead, we handle size and style diversity through novel style interpolation and transfer techniques, which are much more controllable.

Designs must satisfy aesthetic goals. A few works propose to measure one or more aesthetic requirements. Lok *et al.* [15] measure visual balance for automatic layout. Balinsky *et al.* [1] quantify multicomponent document alignment and regularity derived directly from designer knowledge. Ngo *et al.* [18] propose to use fourteen aesthetic characteristics to evaluate interface aesthetics. However, design is too complex to be measured by one or two quantities. It is also unclear whether tens of them are enough nor how to combine these aesthetic quantities together to automate layout.

Layout problems also arise in a number of domains. In text-based documents, templates can be applied without much difficulty as such kind of documents usually follow a linear read-order while banners don't. We refer our readers to a recent review [10] for more discussions in this area. A relatively simpler problem is label layout, in which texts are placed on or around a background image [11, 25]. Boll *et al.* investigate photo album structure, in which photos dominate layout [2]. Sandhaus *et al.* propose to transform a blog into a photo book considering aesthetic requirements [22]. Cao *et al.* [3, 4] try to model several hand-tuned priors for manga panel layout. Gajos and Weld [7] propose a model to specify widget type and position, allowing users to select among different interfaces. Kumar *et al.* [12] present a learning-based system for example-based web page style transfer and size re-targeting by mappings between Document Object Model (DOM) elements of two webpages. Merrell *et al.* [16] and Yu *et al.* [27] optimize their carefully designed energy functions for indoor room layout - furniture placement. Some of the above methods rely on problem-specific properties to work, others tend to use a pair of complex energy function and powerful solver to address the layout problem. However, their methods cannot support multi-size and multi-style design in a steerable way.

Style is a term used to describe the distinctive appearance of a matter, such as speech, image, animation and more. Previous researchers have proposed to interpolate base styles to generate stylistic speech and animation [9, 24]. However, interpolation across different layout styles is dangerous for banner design as the middle of two layouts can easily be non-sense, *e.g.* Figure 4(d). Instead, we propose interpolation to support layout in arbitrary display size rather than to producing new styles.

Figure 2: A typical banner and its elements.

3 THE BANNER LAYOUT PROBLEM

Real banners often comply with rigorous rules to provide a unified visual experience when designing a banner series. In this scenario, texts and images usually appear at similar positions and scales, if taking no account of banner size variation. Layout style learning is to infer such kind of preferences with a collection of training banner exemplars. The banner layout we consider typically consists of a limited number of two-dimensional elements each with an attached role from the following candidates:

Background. A background image can either be a pure-color 2D-canvas or a stylistic wallpaper. All the other elements should be drawn within the background image. We leave the background image as is and optimize the layout of other elements in this paper.

Text. With just a short description of the product, the banner becomes a powerful call to action. The detailed roles of the text can be title, subtitle, etc.

Product. The core of a banner is usually a pretty image that shows off the product or service. Note that the style of the banner is usually determined by the positions of texts and products.

Logo. The logo represents the trademark of the manufacturer or supplier, and its location is usually independent of other elements in the banner.

Tag. The tag element usually positions closely to a product or a text, which is often related to a click-able button.

It should be noted that our method is independent with the exact definition of element roles. Figure 2(a) shows a typical banner, which consists of one background image (Figure 2(b)), four titles (Figures 2(c-f)), one product (Figure 2(g)), one tag (Figure 2(h)) and one logo (Figure 2(i)). By changing the element size and position, it is flexible to produce various styles of banners. However, to handle a variable number of elements and arbitrary element shape, style should be defined with more stable quantities rather than directly measuring each element's position and size.

4 SINGLE SIZE LAYOUT STYLE AND OPTIMIZATION

Similar to previous work, we automate layout design by optimizing an energy function measuring the fitness of a layout style. For this purpose, we first need to decide what should be measured in our style.

Many style indexes, such as the visual balance, unity and symmetry, have been investigated in the literature for the auto layout design [19, 20]. However, these indexes are relatively high level aesthetic quantities and affect layout globally in an unintuitive way. In contrast, we consider the following style indexes closely related to content position and scale:

Margin. Margin reflects arrangement of (one or more) elements relative to banner's boundaries. It is defined as four directional distances from content bounding box edges to the respective boundaries of the banner. Margin of a single element directly constrains its position and size, and is used to characterize large elements such as the product image. Margin of an element group, *e.g.* the group of all texts, regularizes the group's overall position inside the banner. Margin is also related to White space and symmetric balance, which are considered important in previous research [19]. However, we find that directly measuring higher level of aesthetic qualities, such as space and balance, is unnecessary, but inevitably increases complexity.

Relative position. Banners in the same style usually follow the same visual flow or read order, which is another important style indicator. This term is calculated as the position disparity between two correlated elements. Relative position is associated with alignment, which helps define the element anchor during relative position calculation. For example, left-alignment uses the top-left or bottom-left anchor while right-alignment uses the top-right or bottom-right anchor. Alignment is auto-detected during style learning and then fixed. It is a discrete style index which does not need optimization.

Scale. Different styles emphasize different roles by enlarging or shrinking elements of certain roles. Element scale is defined as the height of its corresponding bounding box and the width alters accordingly to keep aspect ratio unchanged. Element scale is typically used for assistant elements such as a tag attached to a product image.

Saliency map [26, 28] is always incorporated into the calculation of the above style indexes in order to capture the real dimension of each element.

Banners of the same style and size probably share some common values and we want to investigate the rule of it. We analyzed thousands of real banners and proposed to use the Gaussian function to model the distribution of each style index. We tested this assumption on various sizes of banners, with more than 50 exemplars for each size and illustrate typical Gaussian fitting results in Figure 3. Our observation indicates that the Gaussian function is surprisingly good for style index distribution. The Gaussian distribution density function for each style can be defined below:

$$g(\Theta_k) = \frac{1}{\sigma_k\sqrt{2\pi}} e^{-(f_k(X)-m_k)^2/2\sigma_k^2}, \tag{1}$$

where X is layout variable (a vector) including positions and sizes of all elements, Θ_k represents the k^{th} style parameter containing m_k

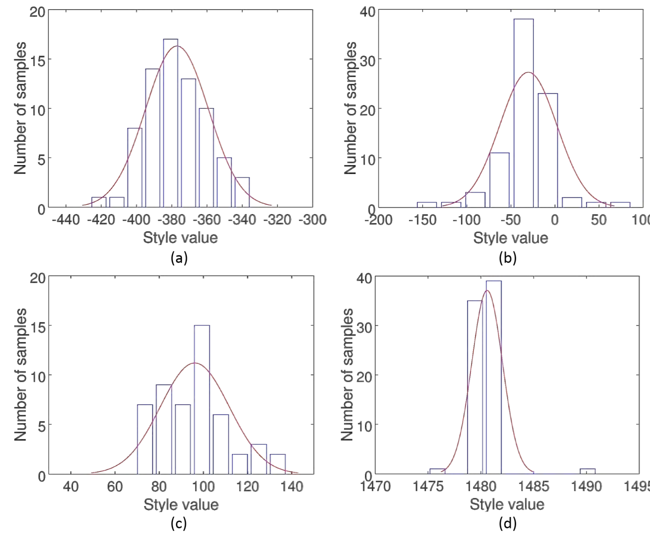

Figure 3: Distributions of four example style indexes with their corresponding fitted Gaussian curves. (a) relative position between the product and the title group for a 720×720 **banner style; (b) right margin of the product image for a** 750×298 **banner style; (c) relative position between the product and the tag for a** 944×348 **banner style; and (d) right margin of the logo for** 2000×500 **banner style.**

and σ_k, which are the mean and standard deviation of the Gaussian distribution for the k^{th} style index respectively, and $f_k(\cdot)$ evaluates the k^{th} style term. A Z-score [8] is defined as $Z_k = \frac{f_k(X)-m_k}{\sigma_k}$, which measures how far the value of the k^{th} style term is from the mean in units of the standard deviation. To measure the overall quality of an input layout design, we can derive an energy function by using the sum of the squared Z-scores as follows:

$$E(X,\Theta) = \sum_{k=1}^{n} Z_k^2 = \sum_{k=1}^{n} \left| \frac{f_k(X) - m_k}{\sigma_k} \right|^2 \tag{2}$$

To be precise, a multi-variable Gaussian function should be controlled by a covariance matrix. However we found that variable-separated Gaussian is good enough in our experiments. Given a set of training banners, one can obtain the style parameter Θ with EM method [17] without difficulty.

Given a learned style parameter Θ, we can minimize the above energy function with global search, *e.g.* parallel tempering [6], refined by local optimization, *e.g.* bounded L-BFGS [29], resulting in a banner layout consistent with its style requirement. Note that the standard deviation is crucial in our energy function to provide elasticity for various number of elements and arbitrary element shape. This is why we need to learn the style parameter from multiple banner exemplars.

In the following two Sections, we extend our style modeling of a specific size for multi-size and multi-style banner design.

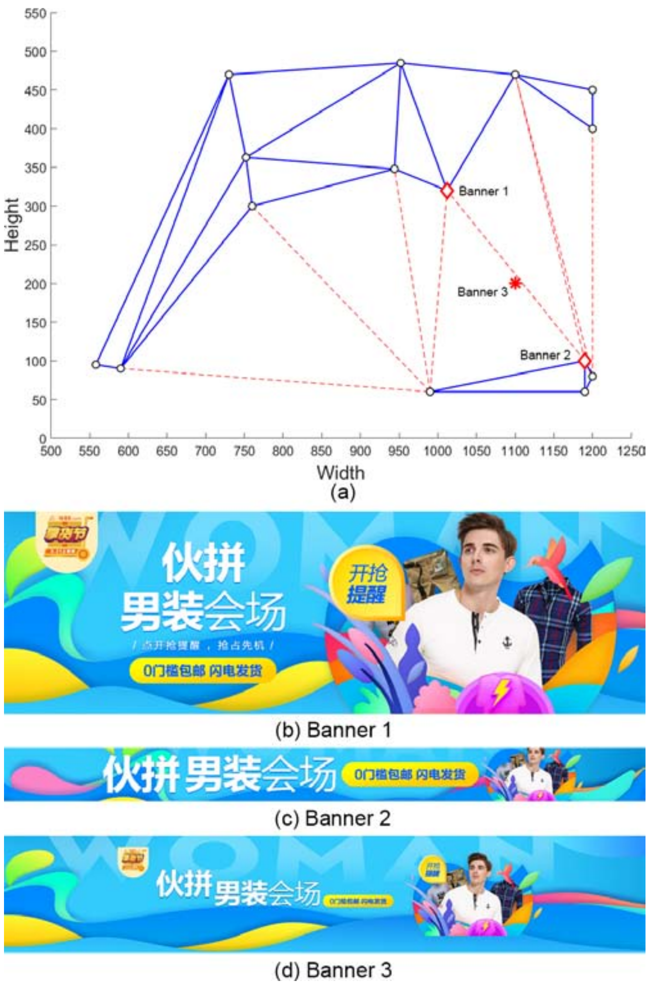

Figure 4: (a) How the discontinuity separates a layout style in different sub-styles (represents as connected components in the graph). Blue solid line indicates its two endpoints are in the same configuration, red dotted line indicates otherwise. (b,c) Two banners of adjacent banner sizes but in different sub-styles respectively; (d) Optimized banner example with the the style parameter from interpolating style parameters of (b) and (c) banner sizes.

5 LAYOUT STYLE INTERPOLATION

To automate banner design of arbitrary size, it is conceptually possible to collect banners of several sizes and learn the style parameter Θ of each banner size, then interpolate trained sizes yielding the style parameter of a requested size and finally optimize the banner with the interpolated style parameter. However, interpolation relies on local smoothness to work but we observed drastic changes in our dataset as shown in Figure 4(a,b). After discussing with professional designers, we realized that such kind of drastic changes are common in multi-size banner design. Note that such kind of discontinuity is difficult to address by previous pure optimization methods. As a result, the key problem here is to correctly detect size discontinuity.

If we plot all sizes of trained banners on a figure, we can see how they scatter in its 2D space (circles in Figure 4(a)). Usually, sizes scatter non-uniformly, but we can use Delaunay triangulation [13] to segment the space into triangles, in each of which one can do triangle interpolation. As we discussed above, not all triangle is safe for interpolation, *e.g.* interpolating banner 1 and banner 2 can easily yield unexpected result shown in Figure 4(d). The reason behind this is that designers usually do not expect a rotated layout just because of moderate size variation. To avoid this, we need to detect large rotation between styles of two banner sizes. Such kind of "rotation" can be measured by directions of relative positions in our style indexes:

$$\alpha = atan2(p_y, p_x) \tag{3}$$

where p_x and p_y are the x and y component of a relative position respectively. In this paper, two style parameters are safe for interpolation when their difference on every relative position direction is less than 45°. If we mark all triangle edges connecting interpolation safe vertices with solid blue line, and other edges with dotted red line, we can usually obtain two or more isolated size regions, in which interpolation is safe. We call the union of all style parameters in each size region a sub-style.

Safe interpolation should not cross two or more sub-styles. To achieve this, we first determine whether the target size locates inside any triangle, *e.g.* in Figure 4(a). If so, relevant sub-styles are those containing one or more vertices of the surrounding triangle. If more than one size regions are found, we do interpolation with each sub-style separately, yielding two or more results for users to select. Banner 3 in Figure 4 is actually such kind of case. If the target size is not inside of any triangle, we seek the nearest edge or vertex to find the relevant sub-style.

There are three possible situations when doing interpolation with a sub-style, as shown in Figure 5. If the target size is inside a triangle of the sub-style (Figure 5(a), we apply Barycentric interpolation [23] directly to the style parameter.

If the target's nearest element is a vertex (Figure 5(c)), we perform uniform content (rather than parameter) scaling so that the interpolated style Θ' of the target banner size D' satisfies:

$$\begin{gathered} S(\Theta', D') = \beta S(\Theta, D) \\ C(\Theta', D')/D' = C(\Theta, D)/D \\ \beta = \begin{cases} \frac{D'_x}{D_x} & \text{if } \frac{D'_x}{D'_y} \leq \frac{D_x}{D_y} \\ \frac{D'_y}{D_y} & \text{otherwise} \end{cases} \end{gathered} \tag{4}$$

where D and Θ are the relevant vertex's banner size and style parameter respectively, S is the content size including width and height of the bounding box spanned by all non-background elements, C is the center position of the bounding box, β is a scalar representing the content scaling ratio between the reference and the target banner size. Content scaling is similar to adjusting the video content when resizing a video player window, in which the content is resized accordingly. The resulting style parameter Θ' is easy to calculate by adjusting the mean of each style index according to Equation 4. Variance of each style index is scaled according to the ratio of banner sizes: $S(D')/S(D)$.

Figure 5: Three interpolation situations. (a) Triangle interpolation: the target size is inside a triangle of the sub-style. (b) Line interpolation: the target size is outside of any triangle of the sub-style and its neighborhood is a line. (c) Vertex interpolation: the target size is outside of any triangle of the sub-style and its neighborhood is a vertex. Target sizes are represented as diamond-cross and is connected with related vertices with red dotted lines. (d-k) Banner examples of corresponding sizes labeled in (a-c) respectively; (j), (h) and (e) are optimized banners with interpolated style parameters from (a), (b) and (c) respectively.

If the target size's nearest element is an edge (Figure 5(b)), we perform the uniform content scaling described above for each edge vertex and average the two styles with a distance weight:

$$\begin{aligned} \Theta' &= w\Theta'_1 + (1-w)\Theta'_2 \\ w &= \frac{dist(D_2, D')}{dist(D_1, D') + dist(D_2, D')} \end{aligned} \tag{5}$$

where D_1 and D_2 are banner dimensions of the first and second edge vertex respectively, Θ_1 and Θ_2 are corresponding style parameters respectively, $dist(\cdot)$ calculates the distance between two banner dimensions.

Interpolation results are illustrated in Figure 5(j,h,e) for each of the three situations described above respectively.

6 LAYOUT STYLE TRANSFER

Style learning and interpolation described in previous two sections successfully model a multi-size style, which can be used to design a banner layout for arbitrary size. This section aims to build the relationship of different styles and facilitate style proliferation.

In real applications, users want to minimize the required number of training banners to learn a new style. The question here is how to transfer from an original multi-size style to a new one according to a few or even only one reference banners. For simplicity, we first consider style transfer with reference banners of a single banner size and discuss how to leverage references of multiple sizes.

Similar to interpolation, style transfer is safe in the sub-style level, so we first need to find the relevant original sub-style. Different from interpolation, we consider rotation in the first quadrant formulated as:

$$\alpha_t = atan2(|p_y|, |p_x|) \tag{6}$$

to support mirrored layout and apply the same (45°) criterion as in interpolation. It should be noted that seeking relevant sub-style may fail according to Equation 6, in which case we cannot apply style transfer.

The original sub-style usually contains a limited number of banner size samples and the reference banner size can be different with any of them. So once the relevant sub-style is found, we interpolate the original sub-style yielding the style parameter of the exact reference banner size. As the motivation of the style transfer indicates, the transfered style should be identical with the reference at least for the reference banner size. This can be easily achieved by modifying the mean of each style index according to the reference banners. For the rest size samples in the original sub-style, we apply non-uniform content scaling to propagate the new style parameter from the reference banner size to other sizes. This non-uniform content scaling satisfies Equation 7:

$$\begin{aligned}\frac{S(\Theta'(D_1), D_1)}{S(\Theta'(D_2), D_2)} &= \frac{S(\Theta(D_1), D_1)}{S(\Theta(D_2), D_2)}\\ \frac{C(\Theta'(D_1), D_1)}{C(\Theta'(D_2), D_2)} &= \frac{C(\Theta(D_1), D_1)}{C(\Theta(D_2), D_2)}\end{aligned} \tag{7}$$

where D_1 and D_2 is any two different banner sizes, $\Theta(D)$ is the original style parameter at banner size D and $\Theta'(D)$ is the new parameter accordingly. S and C share their definitions in the previous section. Equation 7 reflects structure similarity between the original style and the target style, which helps generate style parameter for each banner size. With Equation 7, we can adjust the mean of each style index yielding the transfered style parameter, even with only a single reference banner.

If references of two or more banner sizes can be provided, transfer will be more accurate. In this case, we first apply the non-uniform content scaling for each reference banner size $D'_i, i = 1, 2, ..., m'$ according to Equation 7, yielding multiple candidate style parameters $\Theta'_i(D_j)$ for each original banner size $D_j, j = 1, 2, ..., m$. Then we triangulate the reference banner sizes as in Section 5, seek for triangle, edge or vertex neighborhood of each original banner size following the method described in Section 5, yielding one to three relevant candidate style parameters. Finally, we interpolate them as described in Equation 8:

$$\begin{aligned}\Theta'(D_j) &= w_1\Theta'_1(D_j) + \cdots + w_l\Theta'_l(D_j)\\ w_h &= \frac{dist(D_j, D'_{i_h})}{\sum_{k=1}^{l} dist(D_j, D'_{i_k})}\end{aligned} \tag{8}$$

where l is the number of relevant sizes in reference banners.

7 RESULTS

We implemented our layout modeling, interpolation and transfer algorithms on a PC with Intel Core i7 2.6GHz CPU and 16GB RAM. 1612 well-designed banners were used to train the original multi-size layout style, which was then transfered to a number of distinct styles. Transfered styles can also be used as bases to do transfer again. It cost 10 minutes to learn the 1612 banners but only 1-2 seconds to optimize a new banner and several milliseconds for style interpolation and transfer. Most style learning time was spent on saliency detection.

Figure 6 shows 6 style transfer examples. The first example (a1-a4) vertically flips the relative position between title and product image. The second example (b1-b4) horizontally flips the relative position between title and product image. The third (c1-c4) and fourth (d1-d4) examples not only flip the relative position between title and product, but also re-arrange title lines with different order, alignment and line spaces. The last two examples (e1-f4) illustrate how logo and tag changes before and after style transfer. These results demonstrate that our layout style modeling is a powerful tool for multi-size and multi-style banner design.

It should be noted that some banners are designed with additional constraints out of the core style modeling algorithm discussed above. For example, some designers may prefer logos sticked to the top-left corner of the banner, *e.g.* in Figure 6(f3). Such kind of constraints are not difficult to detect and automatically handled as long as the training or reference banner contains a logo aligned with the top-left corner. For more general cases, we can allow designers to manually add user-specific constraints to control banner optimization results.

Table 1: Received total scores.

Designs	Designer's	Ours with NUCS	Ours with UCS
Total score	95	72	33

To qualitatively evaluate our layout method, we conducted a survey, in which users were shown 20 comparisons among banners generated by professional designers, by our method through style transfer with non-uniform content scaling (NUCS), and by our method with uniform content scaling (UCS). Users were asked to choose the best of the three in each comparison. Two professional designers and 8 amateurs participated in our survey. Statistic results are listed in Table 1. Our method (with NUCS) received comparable credits against professional designs, demonstrating its high layout quality. UCS result received much less credits because it ignores detailed multi-size style structure after style transfer, while NUCS takes it into account as expected.

8 CONCLUSIONS

Automating banner layout in real industry scenarios can be very complex due to drastically varying sizes and rapidly updated styles. This paper addresses this problem with three novel techniques each of which addressing one aspect of this problem. A probabilistic model is firstly introduced with real banner analysis and an energy based optimization is used to automate banner creation for a particular size and style. The style parameter used in the optimization are both size and style dependent. A style parameter interpolation method is then proposed to support arbitrary size banner design based on a small collection of fixed-size style parameters. Finally, a style transfer technique is developed to help generate new multi-size styles.

Experiments show that our method can generate competitive results against professional designs. However, we are still interested in a number of research directions. First, we want to handle training data containing designs of more than one styles. This improvement will greatly help novice users as the current definition of style in our method may not easily align with their intuition. Second, we want to consider element rotation, font types, coloring and interaction with patterned background in our model. This may result in an easy-to-use one-stop solution for banner production. Third, we have tried to solve the banner design problem fully automatically, but there is important applications allowing user interaction deserving more exploration.

Figure 6: Examples of style transfer. The first column shows example banners in original styles, the second column are reference banners representing the target style, the third and fourth columns are optimized banners with transfered style parameters.

REFERENCES

[1] Helen Y Balinsky, Anthony J Wiley, and Matthew C Roberts. 2009. Aesthetic measure of alignment and regularity. In *Proceedings of the 9th ACM symposium on Document engineering*. ACM, 56–65.

[2] Susanne Boll, Philipp Sandhaus, Ansgar Scherp, and Utz Westermann. 2007. Semantics, content, and structure of many for the creation of personal photo albums. In *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 641–650.

[3] Ying Cao, Antoni B Chan, and Rynson WH Lau. 2012. Automatic stylistic manga layout. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 141.

[4] Ying Cao, Rynson WH Lau, and Antoni B Chan. 2014. Look over here: Attention-directing composition of manga elements. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 94.

[5] Steven P Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. 2012. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. In *Design Thinking Research*. Springer, 127–153.

[6] David J Earl and Michael W Deem. 2005. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics* 7, 23 (2005), 3910–3916.

[7] Krzysztof Gajos and Daniel S Weld. 2005. Preference elicitation for interface optimization. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 173–182.

[8] Asghar Ghasemi, Saleh Zahediasl, et al. 2012. Normality tests for statistical analysis: a guide for non-statisticians. *International journal of endocrinology and metabolism* 10, 2 (2012), 486–489.

[9] Keith Grochow, Steven L Martin, Aaron Hertzmann, and Zoran Popović. 2004. Style-based inverse kinematics. In *ACM transactions on graphics (TOG)*, Vol. 23. ACM, 522–531.

[10] Nathan Hurst, Wilmot Li, and Kim Marriott. 2009. Review of automatic document formatting. In *Proceedings of the 9th ACM symposium on Document engineering*. ACM, 99–108.

[11] Ali Jahanian, Jerry Liu, Daniel R Tretter, Qian Lin, Niranjan Damera-Venkata, Eamonn O'Brien-Strain, Seungyon Lee, Jian Fan, and Jan P Allebach. 2012. Automatic design of magazine covers. In *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 83020N–83020N.

[12] Ranjitha Kumar, Jerry O Talton, Salman Ahmad, and Scott R Klemmer. 2011. Bricolage: example-based retargeting for web design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2197–2206.

[13] Der-Tsai Lee and Bruce J Schachter. 1980. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences* 9, 3 (1980), 219–242.

[14] C Karen Liu, Aaron Hertzmann, and Zoran Popović. 2005. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 1071–1081.

[15] Simon Lok, Steven Feiner, and Gary Ngai. 2004. Evaluation of visual balance for automated layout. In *Proceedings of the 9th international conference on Intelligent user interfaces*. ACM, 101–108.

[16] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive furniture layout using interior design guidelines. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 87.

[17] Todd K Moon. 1996. The expectation-maximization algorithm. *IEEE Signal processing magazine* 13, 6 (1996), 47–60.

[18] David Chek Ling Ngo, Lian Seng Teo, and John G Byrne. 2002. Evaluating interface esthetics. *Knowledge and Information Systems* 4, 1 (2002), 46–79.

[19] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning layouts for single-pagegraphic designs. *IEEE transactions on visualization and computer graphics* 20, 8 (2014), 1200–1213.

[20] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. Designscape: Design with interactive layout suggestions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1221–1224.

[21] Lisa Purvis, Steven Harrington, Barry O'Sullivan, and Eugene C Freuder. 2003. Creating personalized documents: an optimization approach. In *Proceedings of the 2003 ACM symposium on Document engineering*. ACM, 68–77.

[22] Philipp Sandhaus, Mohammad Rabbath, Ilja Erbis, and Susanne Boll. 2010. Blog2Book: transforming blogs into photo books employing aesthetic principles. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 1555–1556.

[23] Vaclav Skala. 2008. Barycentric coordinates computation in homogeneous coordinates. *Computers & Graphics* 32, 1 (2008), 120–127.

[24] Makoto Tachibana, Junichi Yamagishi, Takashi Masuko, and Takao Kobayashi. 2005. Speech synthesis with various emotional expressions and speaking styles by style interpolation and morphing. *IEICE transactions on information and systems* 88, 11 (2005), 2484–2491.

[25] Ian Vollick, Daniel Vogel, Maneesh Agrawala, and Aaron Hertzmann. 2007. Specifying label layout style by example. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 221–230.

[26] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2013. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3166–3173.

[27] Lap-Fai Yu, Sai-Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. 2011. Make it home: automatic optimization of furniture arrangement. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 86.

[28] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. 2015. Minimum barrier salient object detection at 80 fps. In *Proceedings of the IEEE International Conference on Computer Vision*. 1404–1412.

[29] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)* 23, 4 (1997), 550–560.