Learning Aesthetics Measure of a Document Page Layout From Designers' Interaction

Ildus Ahmadullin; Niranjan Damera-Venkata

HP Laboratories
HPL-2012-13

Keyword(s):

automated publishing; quantifying aesthetics; designer interaction; probabilistic modeling

Abstract:

Aesthetics evaluation of a document layout is typically performed by a designer. Recently there has been proposed a number of automated systems for document creation. We present a new paradigm to automatically quantify document aesthetics, that can be used in the automated document composition techniques. Given a document template, we specify a few aesthetics parameters that can be adjusted to obtain the most aesthetically pleasing layout of a document page. We use graphical interface, where for a given amount of content designers can adjust the parameters to obtain a layout of the highest aesthetic value. The parameters are stored as a feature vector. After obtaining sufficiently large amount of feature vectors, the feature set is modelled by a Gaussian probability distribution. The resulting model can be used in predicting aesthetic value of a new document, or to compose a document of the highest aesthetic value for a given content.

External Posting Date: January 21, 2012 [Fulltext] Approved for External Publication
Internal Posting Date: January 21, 2012 [Fulltext]

Learning Aesthetics Measure of a Document Page Layout From Designers' Interaction

Abstract

Aesthetics evaluation of a document layout is typically performed by a designer. Recently there has been proposed a number of automated systems for document creation. We present a new paradigm to automatically quantify document aesthetics, that can be used in the automated document composition techniques. Given a document template, we specify a few aesthetics parameters that can be adjusted to obtain the most aesthetically pleasing layout of a document page. We use graphical interface, where for a given amount of content designers can adjust the parameters to obtain a layout of the highest aesthetic value. The parameters are stored as a feature vector. After obtaining sufficiently large amount of feature vectors, the feature set is modelled by a Gaussian probability distribution. The resulting model can be used in predicting aesthetic value of a new document, or to compose a document of the highest aesthetic value for a given content.

CR Categories: I.3.6 [Computing Methodologies]: Computer Graphics—Methodology and Techniques; I.7.4 [Computing Methodologies]: Document and Text Processing—Electronic Publishing;

Keywords: automated publishing, quantifying aesthetics, designer interaction, probabilistic modeling

1 Introduction

Aesthetic document composition is typically considered a hard problem, often requiring professional designer skills. In the traditional publishing industry producing hand-designed single page documents is widely practiced. However, due to high marginal cost of designer services, hand-designed creation of personalized documents is not economically viable. Therefore, automated document layout composition has been the topic that attracted a lot of research. [HURST et al. 2009]

One of the most important concerns in automated document composition systems is producing aesthetically appealing document layouts. It is unrealistic to assume that automatically designed documents can overcome performance of a professional designer. However, some reasonable means to measure aesthetics and distinguish a good layout from an obviously bad one is highly desirable.

One of the most popular works in this area is probably [HARRINGTON et al. 2004]. In this paper the authors study the attributes that degrade the aesthetic quality of a document. Non-linear combination of heuristic measures is proposed to predict the aesthetics of a document. It is claimed that the implementations are relatively efficient and further research in this approach is encouraged.

As opposed to deterministic methods of quantifying document aesthetics, the models can also be designed probabilistically. Each of the document element parameter can be assigned a probability distribution that signifies aesthetics. After combining such distributions one can come up with an overall probabilistic framework of determining aesthetic quality of the layout. [Damera-Venkata et al. 2011] assignes Gaussian prior probability distributions to aesthetics parameters. The distribution parameters are set heuristically. The algorithm uses a probabilistic graphical model to calculate optimal layout parameters, and the aesthetics prior distributions are playing major role in document layout composition.

All of these methods, however lack human designer input. The heuristics in measuring document aesthetics are based on observation of designer prepared layouts. In this paper, we utilize designers' input directly in our algorithm. Once we train the algorithm for a certain template, it can mimic the designers' evaluation in measuring layout aesthetics.

2 Learning from designers' interaction

Document layout can be represented in many different ways. The goal of this paper is to design an automated system that could successfully reproduce designers' aesthetic evaluation of a document page layout. We propose using a graphical interface, where designers can adjust aesthetic parameters of a document layout. Their output is then stored as a feature set and used to fit a probability distribution. This task is not trivial, since designers do not think in terms of a mean and covariance parameters of the parameters they adjust. Allowing them to have a freedom to change many aspects such as relative position and number of different types of document page elements would make our task very difficult. We address this issue by fixing a template and allowing designers to change only a limited number of parameters, and repeat this process for a different document template. Having these constraints it is reasonable to assume that the aesthetics probability distribution can be defined as a Gaussian. Similar approach is used in speech recognition literature, where a speech is represented as a sequence of phonemes or speech segments and each segment is modelled as a Gaussian distribution. [Ostendorf and Roukos 1989]

The authors in [Damera-Venkata et al. 2011] used a concept of a probabilistic page template. An example of such template is illustrated in Fig. 1. The probabilistic template represents only relative positions of page elements, that are dependent on values of probabilistic aesthetics parameters. In description of their algorithm the authors set up prior probabilities on the variables θ_i , modeling them as independent Gaussian random variables. Therefore, for a particular template t, the precision matrix of the variable vector $\boldsymbol{\theta_t} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$ is set to be $\mathbf{\Lambda_t} = diag[\lambda_1, ..., \lambda_5]$, where the entries λ_j are assigned deterministically. However, for an accurate overall document page aesthetics assessment the concept of independency of parameters is not correct, and the precision matrix $\mathbf{\Lambda_t}$ should be full and store covariance factors between different parameters θ_i.

In this paper we propose a method of learning a multivariate Gaus-

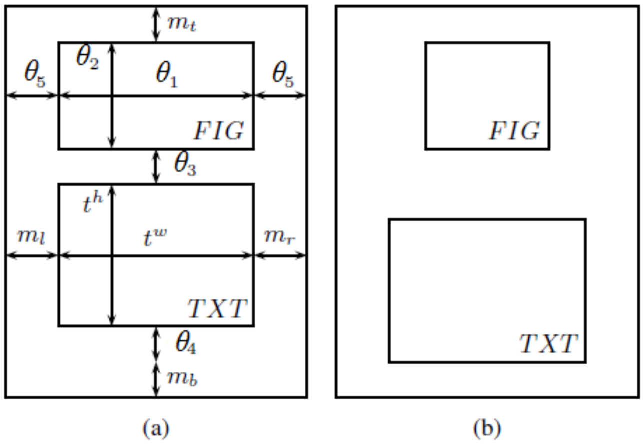

Figure 1: *(a) The template is represented by constant parameters, such as margins, text content height and width; and by variable parameters* $\boldsymbol{\theta_t} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$. *(b) A static template derived by sampling random vector* $\boldsymbol{\theta_t}$.

sian probability distribution of aesthetics parameters $\boldsymbol{\theta_t}$ from designers' input. We initiate the learning step by performing adjustements to the layout for various content amounts. First, we choose a template t from a library of all possible document templates. Then, for the given template, we place a document layout illustration with fixed amount of content on a graphical user interface and allow designers to change the aesthetics parameters. The layout on the GUI changes with every adjustment designer makes for possible values of selected parameters. The images in the layout are represented as green areas and the text regions are filled with dummy text. An illustration of a GUI is shown in Fig. 2. If the designer chooses to see what the actual document looks like for the layout s/he has created, s/he has an option of generating a pdf document. Once all of the variable parameters are set to values that correspond to the best aesthetically pleasing layout for the given content, the designer is asked to give an ultimate rank of the document layout among layouts for all possible variations of content. Designers output is stored as a feature vector $\boldsymbol{\theta_t}$, and a corresponding aesthetic quality rank r_t. In our experiments we let $r_t \in \{1, ..., M\}$ presume integer values from1, that corresponds to the best, to M - the worst layout. Next, we change amount of content and repeat this procedure. Since for every template t we repeat the same process, for the rest of the paper we will drop the template subscript from variables' notations.

3 Model Learning

After finalizing the feature set we proceed with fitting a Gaussian model. A naive way would be to use the ordinary Maximum Likelihood (ML) estimation techniques. However, if during the feature set composition the number of bad quality layout samples was higher than the number of good ones, the result will be seriously biased. The most desirable condition in fitting a Gaussian distribution is to have its values at feature vectors vary relatively to the aesthetic quality ranks. In other words, we would like to have feature vectors that correspond to higher quality layouts closer to the mean, and those that correspond to the poor quality layouts further away. These conditions make the task of calculation Gaussian parameters not trivial. One way to address this problem is to use a linear weighted combination of local ML estimators. In this section we describe the algorithm of determining the weights and Gaussian parameters for the feature set.

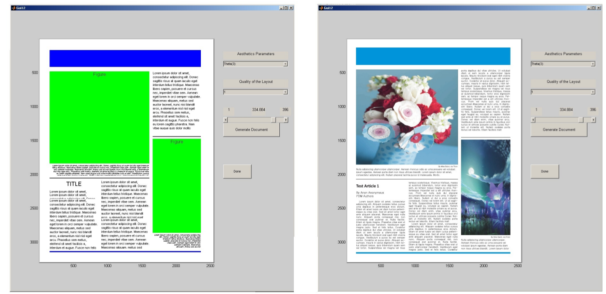

Figure 2: *Implementation of a graphical interface. A designer adjusts the parameters and the corresponding changes are immediately displayed (left) on the GUI. If s/he chooses to see the actual document that corresponds to the given parameters, there is an option of generating a pdf document (right).*

Let $f(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\overline{\boldsymbol{\theta}}, \Sigma)$ denote the Gaussian function with the mean $\overline{\boldsymbol{\theta}}$ and covariance matrix Σ. We denote the class of all feature vectors that correspond to layouts of same quality rank $\Theta_j = \{\boldsymbol{\theta_k} | r_k = j\}$. We denote the whole feature set as $\Theta = \bigcup_{j=1}^{M} \Theta_j$. Then, in the ideal case, we would expect our model to satisfy the following inequalities:

$$f(\boldsymbol{\theta_j}) > f(\boldsymbol{\theta_{j+1}}) \tag{1}$$

, where

$$\begin{gathered} \boldsymbol{\theta_j} \in \Theta_j \\ \boldsymbol{\theta_{j+1}} \in \Theta_{j+1} \\ 1 \leq j \leq M-1 \end{gathered}$$

To achieve this, we require our Gaussian function be such that the feature vectors of the same class are allocated along the same contour. That is we would like the Gaussian parameters to satisfy:

$$\{\overline{\boldsymbol{\theta}}, \Sigma\} = \underset{\overline{\boldsymbol{\theta}}, \Sigma}{\arg\min} \sum_{\boldsymbol{\theta_i} \in \Theta} \left| \frac{1}{N r_i} - f(\boldsymbol{\theta_i}) \right| \tag{2}$$

, where N is the normalization constant of the Gaussian distribution. Note, that in case $r_i = 1$ we obtain $f(\boldsymbol{\theta_i}) \approx \frac{1}{N}$, which means that $\boldsymbol{\theta_i}$ is at the proximity of the mean $\overline{\boldsymbol{\theta}}$. After simple mathematical manipulations we come up with the cost function:

$$\{\overline{\boldsymbol{\theta}}, \Sigma\} = \underset{\overline{\boldsymbol{\theta}}, \Sigma}{\arg\min} \sum_{\boldsymbol{\theta_i} \in \Theta} |2 \log r_i - (\boldsymbol{\theta_i} - \overline{\boldsymbol{\theta}})^T \Sigma^{-1} (\boldsymbol{\theta_i} - \overline{\boldsymbol{\theta}})|^2 \tag{3}$$

In order to obtain the ultimate Gaussian model fit to the whole feature set we begin with fitting Gaussians to the feature vectors of the same class. The mean and the covariance matrix of a local Gaussian fitted to feature vectors of class Θ_j can be calculated as ML estimators:

$$\overline{\boldsymbol{\theta}}_{\boldsymbol{j}} = \frac{1}{|\Theta_j|} \sum_{\boldsymbol{\theta_j} \in \Theta_j} \boldsymbol{\theta_j} \tag{4}$$

$$\Sigma_j = \frac{1}{|\Theta_j|} \sum_{\boldsymbol{\theta_j} \in \Theta_j} (\boldsymbol{\theta_j} - \overline{\boldsymbol{\theta}}_{\boldsymbol{j}})(\boldsymbol{\theta_j} - \overline{\boldsymbol{\theta}}_{\boldsymbol{j}})^T \tag{5}$$

,where $|\Theta_j|$ denotes the cardinality of a class Θ_j. We assign a weight w_j to each of the local Gaussian distribution. Then the overall mean and covariance matrix of the whole feature set can be calculated as:

$$\overline{\boldsymbol{\theta}} = \left(\sum_{j=1}^{M} \sum_{\boldsymbol{\theta_j} \in \Theta_j} w_j \Sigma_j^{-1} \boldsymbol{\theta_j} \right) \left(\sum_{j=1}^{M} w_j \Sigma_j^{-1} |\Theta_j| \right)^{-1} \tag{6}$$

$$\overline{\Sigma} = \sum_{j=1}^{M} \sum_{\boldsymbol{\theta_j} \in \Theta_j} w_j \Sigma_j^{-1} \boldsymbol{\theta_j} \tag{7}$$

After plugging equations (6) and (7) into (3) we come up with a linear system

$$\begin{pmatrix} (\boldsymbol{\theta_1} - \overline{\boldsymbol{\theta}})^T \Sigma_1^{-1} (\boldsymbol{\theta_1} - \overline{\boldsymbol{\theta}}) & \cdots & (\boldsymbol{\theta_1} - \overline{\boldsymbol{\theta}})^T \Sigma_M^{-1} (\boldsymbol{\theta_1} - \overline{\boldsymbol{\theta}}) \\ \vdots & \ddots & \cdots \\ (\boldsymbol{\theta_L} - \overline{\boldsymbol{\theta}})^T \Sigma_1^{-1} (\boldsymbol{\theta_L} - \overline{\boldsymbol{\theta}}) & \cdots & (\boldsymbol{\theta_L} - \overline{\boldsymbol{\theta}})^T \Sigma_M^{-1} (\boldsymbol{\theta_L} - \overline{\boldsymbol{\theta}}) \end{pmatrix} \bullet$$
$$\bullet \begin{pmatrix} w_1 \\ \vdots \\ w_M \end{pmatrix} = \begin{pmatrix} 2 \log r_1 \\ \vdots \\ 2 \log r_M \end{pmatrix} \tag{8}$$

, where $L = |\Theta|$ is the total number of feature vectors in the feature set. Let the RHS of equation (8) be denoted as R, the matrix on the LHS as S and the vector of weights as $\mathbf{w}$. Equation 8 can be solved in the following way:

$$\arg\min_{\mathbf{w}} \|S\mathbf{w} - R\| \tag{9}$$

subject to

$$w_j \geq 0$$
$$1 \leq j \leq M$$

The problem at (9) can easily be solved with the help of techniques described in [LAWSON and HANSON 1974].

The described procedure is quite computationally inexpensive and provide with a weight vector $\mathbf{w}$, that can be plugged in equations (6) - (7) to find the Gaussian parameters. Note that once we have obtained the mean and the covariance matrix of the fitted Gaussian, we can reiterate the procedure and use them to to update the values of the weights w_j. Hence the model training can be summarized as follows:

1. Set initial $\overline{\boldsymbol{\theta}}$.

2. Calulate initial covariance matrices, equation (5).

3. Repeat steps 4-6 until convergence:

4. Solve equation (9) *to obtain the weight vector* $\mathbf{w}$.

5. Update the covariance matrix $\overline{\Sigma}$ *using equation* (7).

6. Update the mean $\overline{\boldsymbol{\theta}}$*using equation* (6).

However, since the feature set was determined by the human input and the quality rankings presume only integer values, it would be rather unrealistic to assume that this algorithm provides with the solution that satisfies the condition we set in (1) for all of the feature vectors without exception. Hence, to handle the outliers we implement Random Sample Consensus (RANSAC) algorithm [FISCHLER and BOLLES 1981]. We randomly divide the feature set into two parts. The first part containing hypothetical inliers, is used to train the model. The second part is used to test againts the fitted model and, if a feature vector is tested well, we also consider it as a hypothetical inlier. If sufficiently many vectors are classified as potential inliers, we reestimate the model from the hypothetical inliers and store the model parameters. We repeat this procedure a number of iterations, and choose the best model from the stored ones.

4 Experimental evaluation

We train our model and compare the output with aesthetics prior model used in [Damera-Venkata et al. 2011]. The intention in experimental evaluation is to demonstrate the superior performance of learned Gaussian parameters compared to the heuristically predetermined parameters which were implemented in [2011]. For this reason we used the same document templates. In order to fully demonstrate the algorithm performace, we selected the templates with different position of landscape and portrait images, spanning one and two columns of a three column document page layout. For each of the templates, we generated 100 samples of aesthetics feature vectors. Each of the feature vectors was given a rank r_j, that ranged from 1 to 5. The feature sets were randomly partitioned into training and testing sets. For each rank r_j, 25% of the corresponding feature vectors were randomly selected and added to the testing set, and the rest were used in the training.

For testing evaluation we calculated the Root Mean Square Error (RMSE) by slightly modifying the cost function (3). In order to visualize the error terms, for each of the feature classes Θ_j in the testing set we calculated the deviation between the actual and estimated ranks:

$$RMSE(\Theta_j) = \sqrt{\sum_{\theta_j \in \Theta_j} \frac{\left| r_j - \exp((\boldsymbol{\theta_i} - \overline{\boldsymbol{\theta}})^T \Sigma^{-1} (\boldsymbol{\theta_i} - \overline{\boldsymbol{\theta}}))/2 \right|^2}{|\Theta_j|}} \tag{10}$$

Aesthetics parameters in the experiments corresponded to the heights of document images, whitespaces in the bottom of the page and above the lower image. The RMSE for each of the classes of the three template feature testing sets are shown in Table 1.

	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5
Template 1	0.3609	0.6892	0.7442	0.6628	1.2401
Template 2	0.2404	0.6691	0.6248	0.7213	1.1230
Template 3	0.1945	0.4640	0.7939	0.8106	1.3046

Table 1: *RMSE values for the feature classes. Each row corresponds to the testing feature set of the templates. Each column corresponds to the classes of ranks 1 through 5. Note that with the definition (10) the RMSE values can be regarded as deviations from the actual integer values of ranks.*

An illustration of the comparison between fitted and predetermined models is shown in Figure 3. Note that since correlations of the aesthetic parameters were not considered in [2011], the corresponding Gaussian has contours that are horizontally inclined. On the other hand, the contours of the fitted Gaussian distribution correctly describe allocation of the feature vectors.

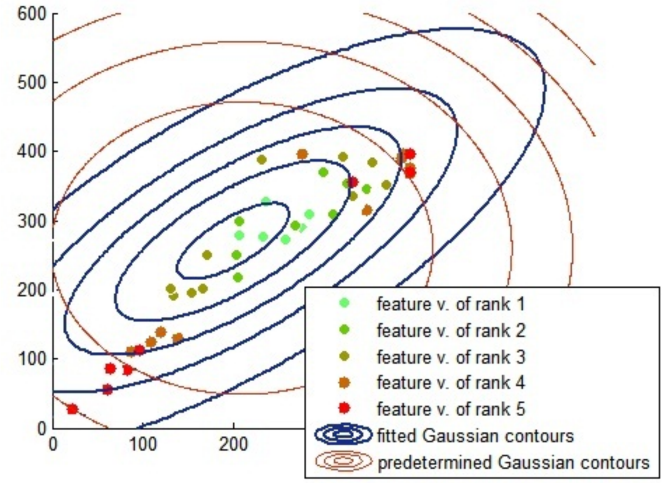

Figure 3: *Comparison between the fitted and the predetermined model. Note that the fitted model correctly estimates the shape of the Gaussian from given training set and distinguishes feature vectors of different ranks. On the contrary, the Gaussian model with the predetermined parameters does not store any correlation between features, and therefore fails to be accurate in document layout aesthetic prediction.*

Samples of document layouts for fitted and predetermined aesthetics models are shown in Fig. 4. Note that the predetermined Gaussian model assigned equal precision factors to the aesthetic parameters that correspond to the heights of document images. Hence, the distinction between portrait and landscape images was not considered. Therefore, in cases of large amount of text content the landscape images suffered from distortion a lot more than the portrait images. This phenomenon is completely eliminated in the learned Gaussian model case.

5 Conclusion

In this paper we described an algorithm of fitting a Gaussian model to aesthetics parameters of a document layout. During data gathering step, the designers are provided with a graphical user interface, where they can adjust document layout parameters as well as give overall aesthetics ranks to the resulting layouts. The feature set is used in training a Gaussian distribution that can be used in automated layout composition described in [2011]. This model stores correlation of the aesthetics parameters in full covariance matrix, which turns out to be very important in document composition. After repeating the learning step for sufficiently large template library, the model can be successfully used in quantifying aesthetic value of single page documents.

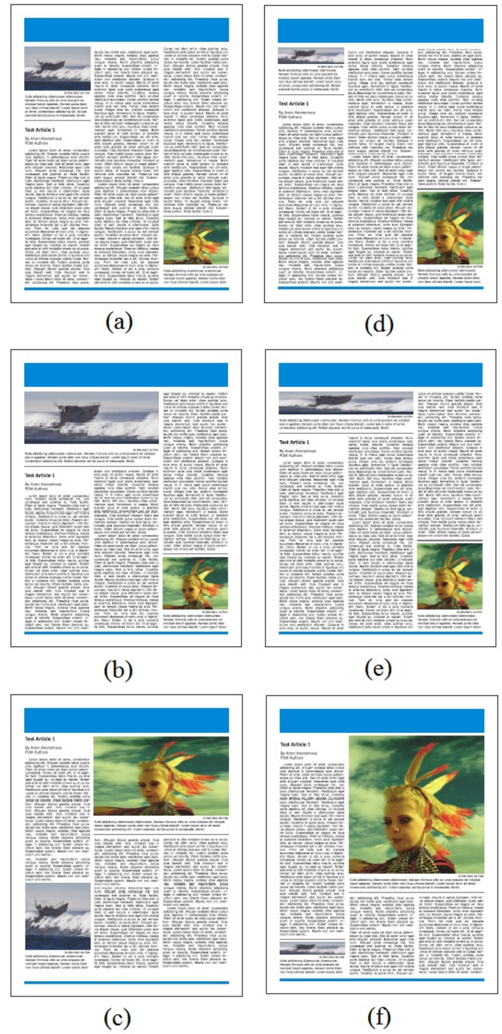

Figure 4: *Documents (a), (b), (c) are the examples of using fitted Gaussian aesthetics model. Documents (d), (e) and (f) are the examples of the predetermined Gaussian model.The advantage of the correlation between image heights is clearly visible in the fitted model examples.*

References

DAMERA-VENKATA, N., BENTO, J., AND O'BIEN-STRAIN, E. 2011. Probabilistic document model. In *Proceedings of the 2011 Symposium on Document Engineering*, ACM, 3–12.

FISCHLER, M. A., AND BOLLES, R. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 381–395.

HARRINGTON, S., NAVEDA, J., JONES, R. P., ROETLING, P., AND THAKKAR, N. 2004. Aesthetic measures for automated document layout. In *Proceedings of the 2004 Symposium on Document Engineering*, ACM, 109–111.

HURST, N., LI, W., AND MARRIOTT, K. 2009. Review of automatic document formatting. In *Proceedings of the 2009 Symposium on Document Engineering*, ACM, 99–108.

LAWSON, C. L., AND HANSON, R. 1974. Solving least squares problems. *The Visual Computer 23*, 161.

OSTENDORF, M., AND ROUKOS, S. 1989. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing 37*, 12 (Dec.), 1857–1869.