Automatic Stylistic Manga Layout

Ying Cao * Antoni B. Chan Rynson W.H. Lau
Department of Computer Science, City University of Kong Kong

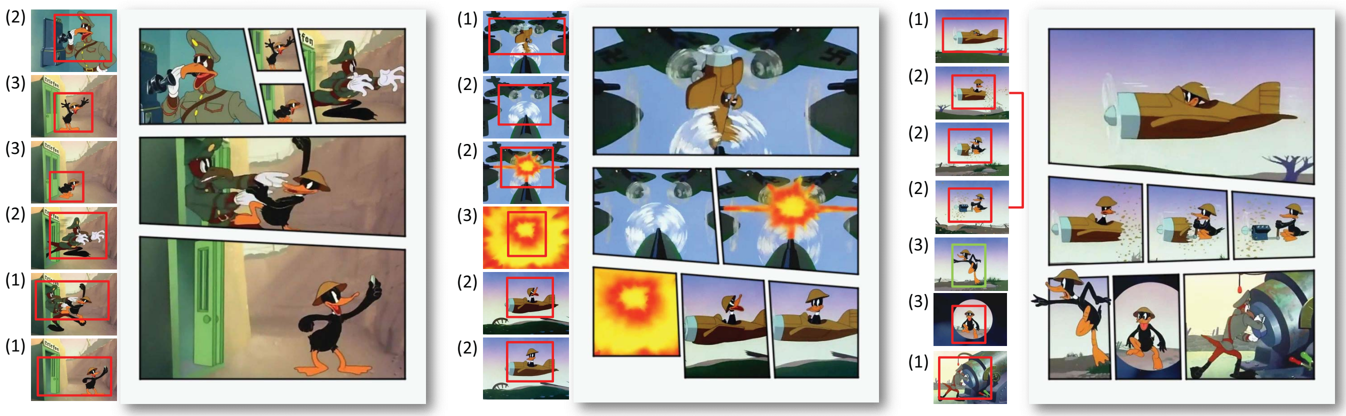

Figure 1: *Three layouts generated by our approach trained on "Fairy Tail". The left side of each example displays the sequence of input artworks ("Daffy: The Commando"(1943) in the public domain), single-panel semantics, including importance-ranking values (within the parenthesis and region of interest (masked by rectangle), as well as optional inter-panel semantics that describe a group of consecutive semantically related panels (grouped by a red line in the rightmost example). The character masked by the green rectangle is chosen for a"fourth wall break" effect. The reading order of each layout is from left to right and then top to bottom.*

Abstract

Manga layout is a core component in manga production, characterized by its unique styles. However, stylistic manga layouts are difficult for novices to produce as it requires hands-on experience and domain knowledge. In this paper, we propose an approach to automatically generate a stylistic manga layout from a set of input artworks with user-specified semantics, thus allowing less-experienced users to create high-quality manga layouts with minimal efforts. We first introduce three parametric style models that encode the unique stylistic aspects of manga layouts, including layout structure, panel importance, and panel shape. Next, we propose a two-stage approach to generate a manga layout: 1) an initial layout is created that best fits the input artworks and layout structure model, according to a generative probabilistic framework; 2) the layout and artwork geometries are jointly refined using an efficient optimization procedure, resulting in a professional-looking manga layout. Through a user study, we demonstrate that our approach enables novice users to easily and quickly produce higher-quality layouts that exhibit realistic manga styles, when compared to a commercially-available manual layout tool.

Keywords: manga, stylistic layout, generative probabilistic model

Links: DL PDF

*yingcao4@student.cityu.edu.hk

1 Introduction

During the last few decades, manga or Japanese comics have gained in popularity across the world due to its unique styles in terms of screening, ballooning and layout. This has resulted in an increasing involvement of the general population in producing their own manga. However, creating high-quality manga typically requires well-trained skill and unique talent. Although various manga production tools (e.g., Manga Studio [MangaStudio 2011]) have emerged to help novices create manga from scratch, none of these tools provide the insight into how manga layout can be done effectively. Manga layout is at the core of manga production. Effective manga layout is utilized by the artists to help storytelling, guide the reader's attention, and enhance the visual attractiveness of manga pages [Rivkah 2006]. However, manga layout is a complicated task; both in-depth understanding and hands-on experience are required to achieve an effective manga layout [Tsai 2002].

In addition to some basic requirements, such as correct reading order and efficient space utilization, Japanese manga artists typically stylize their layout by introducing some customized features, such as: 1) different layout structures (i.e., spatial arrangement of the panels), rather than a single layout template, which augment the visual richness; 2) variations in panel size, with larger panels for important events and smaller panels for scene or moment transitions, which increase the semantic contrast between the panels and make the storytelling more dynamic; 3) irregular panel shapes, instead of uniform rectangles, which make the contents in the panel more dynamic and engaging. These factors characterize manga layout, distinguishing it from the layout of traditional Western comics, which is more rigid and grid-based (e.g., see Figure 2).

Manga artists rely upon their intuition and experience for layout design, which is hard to completely formulate. Hence, previous layout algorithms, based on either heuristic rules [Taniguchi et al. 1997; Thawonmas and Shuda 2008] or energy optimization [Purvis et al. 2003; Cagan et al. 2002; Geigel and Loui 2003; Merrell et al. 2011; Yu et al. 2011], are not applicable, since there are no clear guidelines that can be incorporated into these frameworks. In contrast

Figure 2: *Differences between manga and traditional Western comics. Left: a manga page from "Detective Conan" (©AYOYAMA Gosho / Shogakukan Inc.). Right: a comic strip from "Superman" (SUPERMAN™ and ©DC Comics).*

to this previous work, we propose an approach to produce stylistic manga layout, which is based on flexible style models that are learned from existing manga pages. In essence, our data-driven approach can learn the structure and style of manga layout from examples, thus avoiding the need to define explicit guidelines.

Our approach. To represent the unique stylistic features of manga layout, we introduce three novel models for the spatial arrangement, importance, and shape of the panels in a layout. First, we propose a generative probabilistic model for layout, which can represent the rich and widely varying arrangement of panels on the page. Second, we introduce a panel importance classifier, which ranks the importance of the panel based on its shape. Third, we propose a parametric model for panel shape, which encodes the irregular panel shapes found in manga. The parameters of these style models are learned from pages of a manga series, thus allowing us to encode the style of different manga artists.

Using these three style models, our manga layout algorithm proceeds as follows. For each input artwork, the user specifies an importance ranking and a region of interest. Optionally, the user can also group semantically related panels and mark images for a "fourth wall break" effect, where the character will pop out of the panel. Next, an initial layout is calculated that fits both the input images and the learned generative layout model, via *maximum a posteriori* (MAP) estimation. The initial layout is then refined by jointly optimizing the geometries of the artwork and the layout, while regularizing the panels to fit the learned shape model. Finally, post-processing effects (e.g., the "fourth wall break") are added, and the final layout is rendered. Note that our approach does not require the user to determine any specific attributes of each panel (i.e., the position, size and shape). Instead, the user simply provides a region of interest and importance for each artwork, and the algorithm will automatically generate a professional-looking manga layout.

Contributions. In summary, the main contributions of our paper are as follows:

- We introduce three parametric models for representing the unique stylistic features of manga layout, which are flexible enough to encode the layout styles of various manga series.
- With these style models, we propose an algorithm to automatically and efficiently produce manga layout from input artworks and user-specified semantics.

In addition, we have evaluated the effectiveness of our manga layout tool with a user study, which shows that the layouts by our tool are preferred over those from a commercially-available manual tool. To the best of our knowledge, we are the first to consider a data-driven approach to manga layout, where the unique layout style of a manga series can be learned from its own pages.

2 Related Work

General Layout Problem. The layout problem is a well-studied problem in many engineering fields. A large body of computational methods has been reported for architectural, product, and circuit layouts (see [Cagan et al. 2002; Lok and Feiner 2001] for a survey). For this type of problem, metaheuristic algorithms such as simulated annealing (SA) and the genetic algorithm (GA) are often used to search a large solution space for an optimal layout configuration. Yu et al. [2011] developed a fully-automatic furniture layout system, where the cost function is optimized using SA. However, SA has a high computational cost, while in our setting, we prefer a near real-time performance in order to support interactive refinement. Merrell et al. [2011] generated furniture layouts by sampling a density function with a set of interior design guidelines as its terms. However, such guidelines for manga layout design are not available, and hence the method is not directly applicable. Recently, Merrell et al. [2010] employed real-world data to generate plausible residential layouts. While similar in motivation in adopting a data-driven formulation, our approach handles manga layout whose problem domain is ill-studied and quite different from building layout design. Therefore, their domain-specific approach cannot be easily extended to accommodate our problem.

Comic Layout. Previous works on automatically transforming a sequence of images into comics aim to synthesize comic-like layout [Kurlander et al. 1996; Shamir et al. 2006; Tobita 2010; Durrant et al. 2011]. However, all these methods employ either simple heuristic rules or pre-defined templates, thus limiting their ability to produce rich and distinctive styles. In contrast, our approach takes advantage of styles learned from manga examples to drive layout generation, thereby readily reproducing a variety of manga layout styles. Preu et al. [2007] and Ryu et al. [2008] generated comic pages from a movie, but did not handle the page layout problem, leaving the panels to be arranged manually.

Layout in Video Summarization. Video summarization techniques focus on the visual composition of video frames. Taniguchi et al. [1997] sequentially placed frames, with the goal to effectively utilize space. Uchihashi et al. [1999] and Girgensohn et al. [2003] used a combinatorial-based algorithm to pack keyframes. Calic et al. [2007] optimized a layout of keyframes using dynamic programming, where the optimal solution is a complete layout that best fits a predefined template. Several recent works have attempted to develop various seamless and multi-scale video representation, for the purpose of better video summary and navigation [Goldman et al. 2006; Barnes et al. 2010]. Unlike these methods, our end goal is to achieve a layout that most resembles what manga artists create, rather than a compact or coherent video summary.

Photo or Document Layout. Several methods have been proposed for photo and document layout. Recursive spatial division was used by [Atkins 2004] to produce an adaptive photo layout. A rule-based optimization scheme was adopted in [Geigel and Loui 2003; Purvis et al. 2003]. Jacobs et al. [2003] developed an adaptive grid-based document layout system. Photo collages can also be constructed from photo collections by soft-blending neighboring salient image patches [Rother et al. 2006]. However, none of these techniques aim to reproduce the layout styles unique to manga, which is our focus in this work.

Computational Manga. Computational manga is an emerging research direction in recent years. Several techniques have been developed to facilitate manga creation, such as manga coloring [Qu et al. 2008] and screening [Qu et al. 2006]. Our approach addresses one area of this new research field, namely, manga layout.

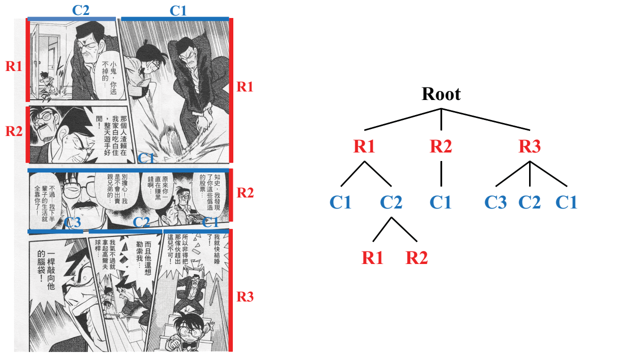

Figure 3: *An illustrative description of recursive spatial division (left) and its division tree (right) for a single manga page from "Detective Conan" (©AYOYAMA Gosho / Shogakukan Inc.). First, the entire page (root node) is split into three rows (red internal nodes R_1, R_2, R_3). Each row is then split into the corresponding columns (blue nodes). Finally, the second column in the first row is further split into two rows (red leaf nodes).*

3 Style Models

Our first challenge is to identify and model the patterns that characterize manga layout. We observe that each manga layout can be stylistically specified by a number of fundamental factors, including the layout structure (i.e., spatial arrangement of panels) and the geometric properties of the panels (i.e., size and shape). In this section, we propose three models for representing these factors, which can be learned from example manga pages.

3.1 Manga Database Creation

As our style models are learned from example manga pages, we first build a database of annotated manga pages. Our database contains approximately 4,000 scanned manga pages from two manga series, "*Fairy Tail*" and "*Detective Conan*", and a Western comic, "*Superman*". The panel vertices in each manga page were labeled manually. In addition, for each manga series, we group the manga pages by the number of panels per page. The style models are learned from each of these groups of pages.

3.2 Generative Model for Layout Structure

When drawing a manga page, artists begin by sketching an initial layout (i.e., layout structure), where the size and position of each panel are roughly determined. To mimic this process, we propose a generative model for layout structure, which can extrapolate the common layout patterns of an artist using a probabilistic model.

Recursive Spatial Division. Our observation is that a basic layout structure can be generated by recursively splitting a page into a number of rows and columns. To describe this generative process, we define two types of spatial division, row division and column division, which correspond to splitting a column into rows, and splitting a row into columns, respectively. In this way, a particular layout structure can be described by a spatial division tree, where each node represents a row or column formed by a series of spatial divisions. An example of such a tree is depicted in Figure 3.

We define the process of splitting a particular row or column as a *spatial division instance* (SDI). Each SDI has a label of the form $\mathcal{L} = Root - R_a - C_b - \cdots - \{R, C\}$, where $Root - R_a - C_b - \cdots$ is the path to the current node in the spatial division tree, and $\{R, C\}$ denotes row or column division. For example, $Root - R_1 - C$ represents the splitting of the first row into columns. Each SDI is associated with a scalar N, which is the number of rows (columns)

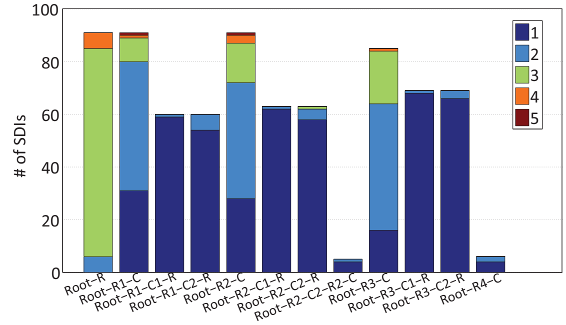

Figure 4: *Distribution of SDIs across 6-panel pages from "Fairy Tail". Each bar corresponds to a particular SDI, with colors indicating the number of rows/columns formed.*

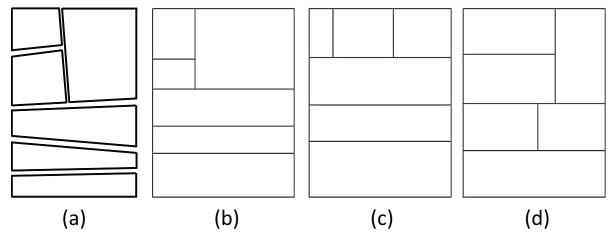

Figure 5: *Layout structures sampled from our generative layout model. Note that our generative model both reproduces layout structure (b) of a training example (a) and extrapolates stylistically plausible variants ((c) and (d)), which are similar to (b) but not in the training set.*

resulting from the split, and a N-dimensional vector $\mathbf{X}$, which is the splitting configuration, i.e., the normalized heights (widths) of the resulting rows (columns). The distribution of SDIs across manga pages generally indicates the preference of the artist in drafting a layout. For example, Figure 4 shows the distribution of SDIs across 6-panel pages from "*Fairy Tail*". In this case, the distribution of N for $Root - R$ suggests that the artist favors layout structure with three rows when arranging 6 panels on one page.

Generative Model. We next propose a generative model for layout structures, which is based on the probability distribution of SDIs and the generative process discussed above. Starting with $\mathcal{L} = Root - R$, we first sample N (the number of rows or columns formed by splitting) from a probability distribution $P(N|\mathcal{L})$, which represents the frequency that a particular N occurs for each $\mathcal{L}$. Next, given N and $\mathcal{L}$, we sample the splitting configuration $\mathbf{X}$, which determines the geometry of each row or column resulting from the split, from a Dirichlet distribution $P(\mathbf{X}|N, \mathcal{L})$. The Dirichlet distribution is defined over the unit $(K-1)$-simplex, i.e., $\{(x_1, \ldots, x_K)|x_i \in [0,1], \sum_i^K x_i = 1\}$. Therefore, a splitting configuration can be naturally regarded as a sample from this distribution. The process repeats recursively for each new $\mathcal{L}$ generated from the splitting process, until the desired number of panels is reached. The parameters of the distributions are estimated from a set of labeled manga pages, thus allowing the model to learn the layout structures of various manga artists. In contrast to [Calic et al. 2007], which uses a fixed number of layout templates, our generative model can extrapolate an infinite number of plausible layout structures, thus improving the richness of the possible layouts. Figure 5 shows several layout structures sampled from our generative model that was trained on 6-panel pages from "*Fairy Tail*".

Parameter Estimation. The parameters of the generative model are learned from a set of labeled manga pages, from which the SDIs are automatically obtained (see supplementary). For each SDI of label $\mathcal{L}_i$, we extract the corresponding number of rows/columns

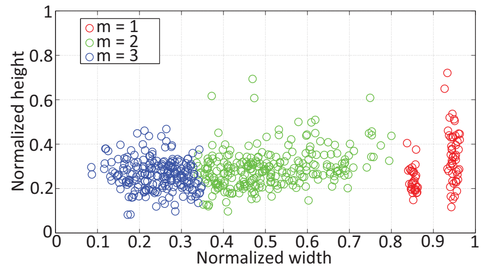

Figure 6: *Panel clustering in 2D feature space. All the panels are extracted from 6-panel manga pages, and are clustered into 3 classes. Each circle denotes an individual panel, and its color indicates cluster membership.*

N_i, and the splitting configuration $\mathbf{X}_i$, resulting in a set of SDI samples $\{N_i, \mathbf{X}_i\}$. Then, $P(N|\mathcal{L}_i)$ is estimated using the relative frequency of $\{N_i\}$, and for each N_i, the Dirichlet parameter vector $\boldsymbol{\alpha}$ of $P(\mathbf{X}|N_i, \mathcal{L}_i)$ is learned using maximum likelihood estimation [Naryanan 1991] on the corresponding $\{\mathbf{X}_i\}$. When only a few SDI samples are available for a particular N_i, we set $\boldsymbol{\alpha} = 1$, which reduces to a uniform distribution over splitting configurations.

3.3 Panel Importance

In manga layout, the sizes of the panels vary throughout the page, with larger panels typically for more important events. To represent this concept, we define the panel *importance*, denoted as $m \in \{1, 2, 3\}$, which indicates the importance ranking of a panel, with $m = 1$ being the most important (largest), and $m = 3$ being the least important (smallest).

To automatically discover the panel importance present in a manga dataset, we use unsupervised clustering to analyze the geometric properties of the panels. As the panel size also depends on the number of panels on the page, we separately analyze each group of layouts with the same number of panels. First, we represent each panel with a 2D feature (nw, nh), where nw and nh are the width and height of the panel's bounding box, normalized w.r.t the page width and height. Next, we group the panels into 5 clusters using a Gaussian mixture model trained with the expectation maximization algorithm [Bishop 2006]. The clusters are then sorted in descending order by the L_2 norm of the cluster centers. Assuming that there are more panels with normal importance ($m = 2$) than those that are of most or least importance ($m = 1$ or $m = 3$), we finally designate the first cluster (largest panels) and last cluster (smallest panels) as $m = 1$ and $m = 3$, while grouping all intermediate clusters into the cluster of $m = 2$.

An example panel clustering appears in Figure 6. The clustering algorithm finds that the panel width consistently increases with its importance, while the variations in height are similar across all importance ranks. This suggests a relationship between the importance and shape of the panels in existing manga pages. In particular, important panels are wide rectangles that fill the entire row of the page, whereas less important panels are vertical rectangles, which leave space for other panels in the row. Finally, we define a panel importance classifier, which takes the normalized width and height of an input panel and outputs the importance of its closest cluster center. The variation of the panel shape for each importance cluster is learned by fitting a parametric shape model, as discussed next.

3.4 Panel Shape Variation Model

In contrast to western comics, panels in manga layouts are characterized by their irregular shapes. To capture this shape variability, we exploit a method analogous to the active shape model (ASM) [Cootes et al. 1995]. Given a set of panels with a particular importance, we first perform Procrustes analysis to align them into a common coordinate system, and then describe each panel by a shape vector $s_i = (x_1, y_1, ..., x_4, y_4)^T \in \mathbb{R}^8$, where (x_j, y_j) is the coordinate of j-th vertex, normalized w.r.t. the page width and height. Using principal component analysis and retaining the top 3 principal components, we build a shape variation model as:

$$s = \bar{s} + \mathbf{U}\boldsymbol{\beta}, \quad (1)$$

where $\bar{s} \in \mathbb{R}^8$ is the mean of all shape vectors, $\mathbf{U} \in \mathbb{R}^{8\times3}$ is the eigenvector matrix, and $\boldsymbol{\beta} \in \mathbb{R}^3$ is the shape parameter vector. For a new shape s_{new}, the best fitting shape allowable by the model is given by $\boldsymbol{\beta} = \mathbf{U}^T(s_{new} - \bar{s})$. To ensure that panel shapes deform in a reasonable way, we enforce constraints on the parameter vector,

$$-\sqrt{\lambda_i} \le \beta_i \le \sqrt{\lambda_i}, \quad i = 1, 2, 3, \quad (2)$$

where β_i is the i-th parameter in $\boldsymbol{\beta}$, and λ_i is the eigenvalue associated with the i-th eigenvector. The shape variation model will be used in both the constrained editing UI of the layout tool, and in the final layout optimization to regularize the shapes of the panels.

4 Manga Layout Approach

Figure 7 presents an overview of our approach, which consists of two main components, offline learning of the style models and online layout generation. In the offline stage, the style models are learned from a training set of labeled manga pages, as discussed in Section 3. In the online stage, given a set of input artworks, the user begins by specifying the semantics, including the region of interest (red or green rectangle on top of each artwork), importance-ranking (the numbers) as well as semantically related panels (connected by a red line), via our constrained editing UI. Next, an initial layout is found that best fits the the user-specified semantics and the generative layout model, using MAP estimation. The final layout is calculated by solving an energy minimization problem using an efficient alternating optimizer that is regularized by the learned shape variation model. The automatically generated layout can be further edited by the user, with the optimizer re-solving for the final layout while treating the user's changes as additional constraints.

One key advantage of our layout algorithm is its flexibility, with the style of the layouts produced by the algorithm depending entirely on the training set used to learn the style models in the offline stage. By using different training sets, we can reproduce the layout styles specific to different manga series (see results in Section 5).

4.1 Constrained Editing Interface

We provide a constrained editing interface, which allows the user to rapidly specify pertinent semantics of the input artworks, including importance, region of interest (ROI), and inter-panel relationship. A demonstration of our interface appears in the accompanying video.

Given a sequence of input artworks, the user first selects the number of panels N in one page. Then, for each artwork, the user assigns a panel importance value (Section 3.3) to indicate its importance ranking, and selects the ROI by dragging a *deformable rectangular mask* (DRM) in the artwork. Our interface will automatically center the DRM around the most salient region in the artwork. To achieve this, we first generate a saliency map by taking the product of a low-level saliency map by [Itti et al. 1998] and high-level face detection map by [Viola and Jones 2004]. Next, the optimal window center and scale are determined by finding the window in the saliency map with the largest Gaussian-weighted sum of saliency values.

The user can manipulate the DRM by translating or deforming this mask, in order to accurately select the intended ROI. In this step,

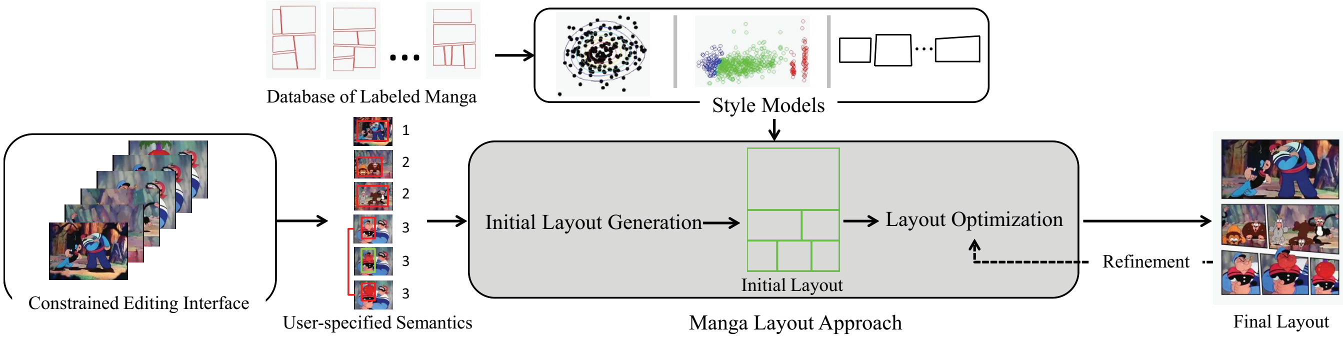

Figure 7: *Overview of the proposed approach.*

we aim to offer the user the flexibility to freely choose any region, as well as useful suggestions about what field-of-view an artist is likely to use to capture interesting objects. To achieve this, the geometry of the DRM is allowed to be scaled and translated arbitrarily, but only deformed within the allowable deformation space of the panel shape variation model of the selected *importance*. In order to ensure the DRM to be rectangular, the panels are represented by their bounding boxes when used to learn the deformation model. Once the ROI is selected for an artwork, we treat its shape as the *image geometry* of the artwork in subsequent processing.

In addition, we also allow the user to optionally group semantically related panels by drawing a line across a set of panels. In this case, our algorithm will take into account inter-panel semantics during the initial layout generation step as will be discussed next.

4.2 Initial Layout Generation

Given the input artworks and the user-specified semantics, the first step of our layout algorithm is to generate an initial layout. On the one hand, the initial layout should exhibit the style expressed by a manga artist, while on the other hand, it should adapt well to the user-specified semantics. One simple solution is to retrieve the best-fitting layout from our database of labeled manga pages, by evaluating the compatibility between the input artworks and the existing layouts. However, since we allow the user to specify an arbitrary *importance* value and *image geometry* for each artwork, direct retrieval will fail to find a proper initial layout when the user's specifications do not well match any of the labeled manga pages (Figure 8(b)). This may lead to an unsatisfying layout, as exemplified in Figure 8(c), where more important panels are erroneously made smaller than less important ones.

Instead of using a fixed database of layouts, we employ a generative probabilistic framework, where our generative model from Section 3.2 serves as the prior distribution over possible initial layouts. Our goal is to find the optimal initial layout that fits both the generative model and the user-specified semantics. In essence, our generative model has the ability to extrapolate an infinite number of valid initial layouts, which conform to the learned layout style.

Formally, let L be the initial layout and I be input artworks with user-specified semantics. We assume a prior distribution on the layout, $L \sim p(L)$, which takes the form of the generative layout model. The prior distribution essentially scores all possible layouts, with higher scores given to layouts that exhibit characteristics similar to those in the training set. The problem of finding the best layout L to a given I can now be solved using MAP estimation,

$$L^* = \arg\max_L p(L|I) = \arg\max_L \underbrace{\log p(I|L)}_{\textit{likelihood term}} + \underbrace{\log p(L)}_{\textit{prior term}}. \quad (3)$$

The prior term is computed directly from the generative model, $p(L) \propto \prod_{i=1}^{r} p(\mathbf{X}_i|N_i, \mathcal{L}_i)p(N_i|\mathcal{L}_i)$. The likelihood term $p(I|L)$ measures how well the initial layout matches the input artworks and user-semantics. In our formulation, there are two types of semantics, i.e., single-panel semantics (*importance* and *image geometry*), and inter-panel semantics (groups of semantically related panels). Hence, the likelihood term consists of two independent terms:

$$\log p(I|L) \propto \omega E_{single}(I, L) + (1-\omega)E_{inter}(I, L). \quad (4)$$

where $E_{single}(I, L)$ and $E_{inter}(I, L)$ are the single-panel and inter-panel likelihood terms, respectively. ω is a weighting parameter that is set to 0.5 in our implementation.

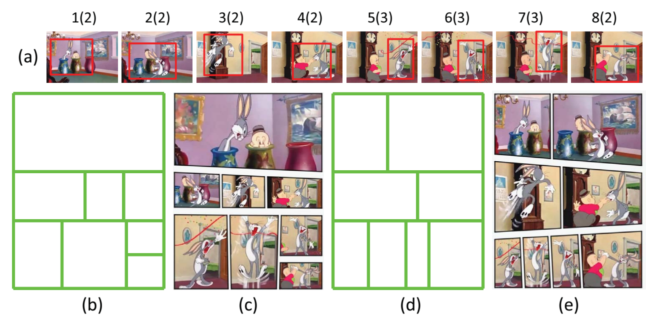

Figure 8: *Comparison to simple retrieval method for initial layout generation. (a) Input artworks ("Bugs Bunny: The Wabbit Who Came to Supper"(1942) in the public domain). (b) Best initial layout retrieved by matching the input artworks with the candidates from our database of labeled manga pages, using the single-panel likelihood function in Section 4.2. (c) Final layout generated based on (b). (d) Best initial layout by our model-based approach. (e) Final layout generated based on (d).*

Single-panel Likelihood Term. $E_{single}(I, L)$ measures the fitness between the input artworks and the layout. Assuming conditional independence, we define $E_{single}(I, L) = \log \prod_{j=1}^{n} p(I_j|P_j)$, where I_j and P_j are the j-th artwork and its panel, respectively. Let M_{I_j} be the user-specified *importance* of the j-th artwork, and M_{P_j} be the *importance* of corresponding panel determined by panel importance classifier (Section 3.3). We define $p(I_j|P_j)$ as:

$$p(I_j|P_j) \propto \begin{cases} \epsilon, & \text{if } M_{I_j} \in \{1,3\} \text{ and } M_{I_j} \neq M_{P_j} \\ e^{\frac{-1}{2\sigma^2}\|\mathbf{x}_{I_j} - \mathbf{x}_{P_j}\|^2}, & \text{otherwise} \end{cases} \quad (5)$$

where ϵ is a small value (set to e^{-10} in our implementation), and $\mathbf{x}_{I_j}$ and $\mathbf{x}_{P_j}$ are the geometric feature vectors of I_j and P_j. Although a wide range of geometric features (e.g., aspect ratio, normalized pixel area, etc.) could be used, we have found that the aspect ratio alone works well in practice. Note that the ϵ term will greatly penalize the layouts where the perceptual contrast (relative size) between the smallest and largest panels is small, thus suppressing the layouts that are likely to be functionally erroneous.

Inter-panel Likelihood Term. Considering the semantic relationship between panels (e.g., panel-to-panel transition) may improve the storytelling ability of the generated layouts [McCloud 1994; McCloud 2006]. The principle is to arrange consecutive panels, which are semantically related, along a smooth path, such that the readers can mentally assemble these fragments as a whole easily. Furthermore, it is also advantageous to have semantically related panels span the entire row or column, in order to align the shot transitions with the row or column boundary. Therefore, we allow the user to optionally specify inter-panel semantics by grouping several consecutive panels, and design the inter-panel likelihood term $E_{inter}(I, L)$ to accommodate such constraints.

Formally, let $\mathcal{S} = \{\mathcal{G}\}$ be set of panel groups selected by the user. We formulate the inter-panel likelihood term as:

$$E_{inter}(I, L) = \log \sum_{\mathcal{G}\in\mathcal{S}} \frac{E_s(\mathcal{G}, L) + E_p(\mathcal{G}, L)}{2|\mathcal{S}|}, \qquad (6)$$

where $E_s(\mathcal{G}, L)$ measures the smoothness of the path among the panels, and $E_p(\mathcal{G}, L)$ encourages the panel group to span the entire row or column, thus aligning shot transitions with the row or column boundaries. For a group $\mathcal{G}$, let $\mathbf{p}_k$, $k = 1, \cdots, |\mathcal{G}| - 1$, be the vector (path) from the center of the k-th panel to the center of the $(k + 1)$-th panel in the group. We define $E_s(\mathcal{G}, L)$ as,

$$E_s(\mathcal{G}, L) = \begin{cases} \frac{\cos(4\theta(\mathbf{p}_k,\mathbf{e}_1))+1}{2}, & |\mathcal{G}| = 2 \\ \frac{1}{|\mathcal{G}|-1} \sum_{k=1}^{|\mathcal{G}|-2} \frac{\cos(\theta(\mathbf{p}_k,\mathbf{p}_{k+1}))+1}{2}, & |\mathcal{G}| > 2 \end{cases}. \qquad (7)$$

When there are two panels in the group, $\theta(\mathbf{p}_k, \mathbf{e}_1)$ is the angle between $\mathbf{p}_k$ and the horizontal axis $\mathbf{e}_1$, and the term encourages the two panels to align horizontally or vertically. When there are more than 2 panels, $\theta(\mathbf{p}_k, \mathbf{p}_{k+1})$ is the angle between consecutive paths $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$, and the term measures the smoothness of the entire path, with the highest value obtained when all the panels are collinear. Finally, $E_p(\mathcal{G}, L)$ is a penalty term, which equals 1 when the first panel in $\mathcal{G}$ is at the beginning of a row/column and the last panel is at the end of a row/column, and 0 otherwise.

Optimal Initial Layout Estimation. There is no closed-form solution to the MAP optimization problem in (3). Although a solution could be found using iterative algorithms, such as deterministic annealing [Geman and Geman 1984] or variational approximations [Bishop 2006], they are computationally expensive and thus unsuitable for an interactive application. Instead, we solve (3) by sampling a set of layouts $\mathcal{O}$ from our generative model, $L \sim p(L)$, and selecting the one that maximizes the *MAP score*, $\log p(I|L) + \log p(L)$. In general, this strategy will not find the global optimum exactly, but will give a close solution when there are enough samples. This is sufficient for our purpose, since in this stage we only need an initial layout to roughly specify the spatial arrangement of the artworks; the initial layout will be further refined in the next stage. To achieve a tradeoff between computational efficiency and solution quality, we empirically set $|\mathcal{O}|$ to 500. Furthermore, instead of maintaining a fixed set of sampled layouts, we sample a new collection of layouts whenever our algorithm is run. This dynamic updating improves the variability of the initial layouts, and allows us to explore the full sample space.

Figure 8 exemplifies the advantage of our model-based approach over a simple retrieval-based method. Due to the diverse set of candidates extrapolated by the generative process, our method is more likely to produce a higher-quality initial layout, and thus a better final layout that is functionally correct and visually pleasing.

4.3 Layout Optimization

Given the initial layout, the goal is to produce a final layout where the images fit well into their corresponding panels, while reproducing the variability in panel shapes, which is a characteristic of manga. We formulate this refinement procedure as a joint optimization problem over the layout mesh and image geometries.

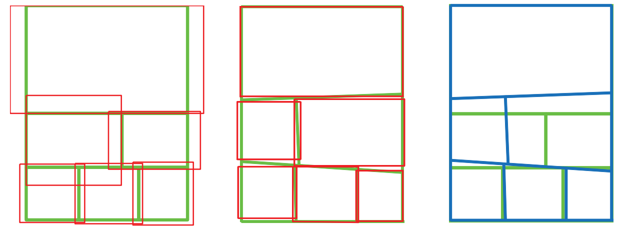

Figure 9: *Layout optimization example. (a) Initial placement of the image geometries (red) and layout mesh (green). (b) Configuration after optimization. (c) Deformed layout mesh after optimization (blue) and the initial layout mesh (green).*

Let $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_m)^T$ be a $2m$-dimensional vector of vertex positions of the layout mesh, where $\mathbf{v}_i$ is 2D coordinate vector and m is the number of the vertices. The images are initially placed in the layout by centering each image in its panel, and enlarging the image to completely cover the panel (if necessary), e.g., see Figure 9 (a). Next, this initial arrangement is further refined to fulfill our objective above. Assuming that each image geometry will undergo a geometric transformation T_i, we formulate the refinement procedure as an optimization problem over $\mathbf{V}$ and $\mathcal{T} = \{T_i\}_{i=1}^n$, and develop an efficient solver using a two-step alternating minimization scheme, which rapidly converges to the global optimum. Finally, the optimal $\mathbf{V}$ and $\mathcal{T}$ are used to render the final layout. Figure 9 illustrates our layout optimization.

Energy Function. We denote the i-th panel geometry in the layout mesh by its four vertices $\{\mathbf{v}_{\gamma(i,j)}\}_{j=1}^4$ where γ is a function to index the j-th vertex of the i-th panel in $\mathbf{V}$, and the image geometry of the artwork associated with i-th panel by its four corners $\{\mathbf{u}_{ij}\}_{j=1}^4$. Furthermore, to maintain the aspect ratio of the artwork, we assume each image geometry can only undergo a rigid transformation, composed of a uniform scaling and a translation, i.e., $T_i = \{s_i, \mathbf{t}_i\}$, where s_i is a scaling factor and $\mathbf{t}_i$ a translation vector. We measure the fitness between the i-th image geometry and its corresponding panel via a discrepancy energy:

$$\mathbf{D}_i = w_i \sum_{j=1}^{4} \|\mathbf{v}_{\gamma(i,j)} - (s_i\mathbf{u}_{ij} + \mathbf{t}_i)\|^2 + \lambda\|\mathbf{t}_i\|^2, \qquad (8)$$

where $\|\mathbf{t}_i\|^2$ is a regularization term weighted by λ (empirically set to 3). Notice that this energy formulation will encourage the image geometry to lay around the center of the panel geometry, which implicitly encodes the constraint that salient objects should appear in the central part of the panel in manga [McCloud 2006]. We also include a weighting coefficient w_i on each energy term, in order to further constrain the transformation of image geometries. In particular, we use $w_i = \frac{1}{1+exp(-t)}$, where t is the ratio of the pixel area of detected faces to that of the image geometry, and normalize the weights to sum to 1. An image with a large face region will have a higher weight, and therefore a higher penalty on its transformation, thereby well preserving the face region in the panel.

The total energy function is defined by summing the individual discrepancy energy terms,

$$\mathbf{E} = \alpha \sum_{i=1}^{n} \mathbf{D}_i + \beta\|\mathbf{V} - \mathbf{V}_0\|^2, \qquad (9)$$

where the second term keeps the layout from deforming too far from the initial layout $\mathbf{V}_0$, and α and β are weights (empirically set to 10 and 1, respectively).

Boundary and Collinearity Constraints. We introduce a set of constraints on the layout vertices in order to achieve valid results in the context of manga layout. To guarantee that a rectangular layout mesh is obtained after optimization, we constrain the vertices at the four corners of the page to be fixed, and the vertices on the boundaries of the page to only slide along their respective boundaries. Formally, let $\mathbf{v}^0_{\gamma(i,j)}$ be the initial value of $\mathbf{v}_{\gamma(i,j)}$. The boundary constraints are written as:

$$\mathbf{v}_{\gamma(i,j)} = \mathbf{v}^0_{\gamma(i,j)}, \quad \text{if } \mathbf{v}^0_{\gamma(i,j)} \text{ is at a page corner },$$

$$\mathbf{v}_{\gamma(i,j),x} = \begin{cases} 1 & \text{if } \mathbf{v}^0_{\gamma(i,j)} \text{ is on the left boundary} \\ w & \text{if } \mathbf{v}^0_{\gamma(i,j)} \text{ is on the right boundary} \end{cases}, \tag{10}$$

$$\mathbf{v}_{\gamma(i,j),y} = \begin{cases} 1 & \text{if } \mathbf{v}^0_{\gamma(i,j)} \text{ is on the top boundary} \\ h & \text{if } \mathbf{v}^0_{\gamma(i,j)} \text{ is on the bottom boundary} \end{cases}.$$

The boundary constraints can be written compactly as a set of linear equality constraints on $\mathbf{V}$,

$$\mathbf{m}_i^T \mathbf{V} = b_i, i = 1, \ldots, l, \tag{11}$$

where $b_i \in \{1, w, h\}$. $\mathbf{m}_i \in \mathbb{R}^{2m}$ is a vector whose element equals 1 if $V_i \in \{1, w, h\}$, and 0 otherwise.

In most manga layouts, the edges of neighboring panels in the same row or column are typically aligned together. Thus, we introduce a collinearity constraint, by imposing that two collinear edges in the initial layout should still be collinear after optimization. While it is feasible to devise an analytic formulation of collinearity to directly incorporate into our optimization framework, the collinearity constraints must be expressed as quadratic equality constraints, which precludes convexity of the optimization problem. Instead, we enforce the collinearity conditions by directly modifying the layout geometry $\mathbf{V}$. Specifically, vertices that should be collinear in the layout mesh are projected onto their best fitting straight line. We have found that this simple strategy works well in practice. Compared with directly introducing quadratic constraints, our strategy allows us to derive an efficient alternating solver, which yields high-quality solutions and fast convergence.

Alternating Solver. With the energy function and the boundary constraints defined above, the optimal $\mathcal{T} = \{T_i\}_{i=1}^n$ and $\mathbf{V}$ can be found by solving a constrained optimization problem,

$$\{\hat{\mathcal{T}}, \hat{\mathbf{V}}\} = \arg\min_{\mathcal{T},\mathbf{V}} \mathbf{E}, \ \text{ s.t. } \ \mathbf{m}_i^T \mathbf{V} = b_i, \ \ i = 1, \ldots, l. \tag{12}$$

Since both $\mathcal{T}$ and $\mathbf{V}$ are unknown and correlated with each other, we find their optimal values using an alternating minimization scheme, which alternates between the estimation of $\mathcal{T}$ with $\mathbf{V}$ known, and the estimation of $\mathbf{V}$ given $\mathcal{T}$, until convergence.

Our alternating solver begins by estimating the spatial transformation $T_i = \{s_i, \mathbf{t}_i\}$ for each image geometry, while assuming that the layout $\mathbf{V}$ is known. Note that when $\mathbf{V}$ is a constant, the individual discrepancy energy terms are independent of each other. Hence, $\{T_i\}$ can be be solved separately in a standard least-squares manner.

Next, we assume $\mathcal{T}$ is known and calculate the optimal $\mathbf{V}$. Rewriting (9) as a function of $\mathbf{V}$ and ignoring additive constants, we solve for $\mathbf{V}$ by minimizing $\mathbf{E}$ with the boundary constraints in (11),

$$\begin{aligned} \hat{\mathbf{V}} &= \arg\min_{\mathbf{V}} \alpha\|\mathbf{A}\mathbf{V} - \mathbf{c}\|^2 + \beta\|\mathbf{V} - \mathbf{V}_0\|^2 \\ &\text{s.t. } \ \mathbf{m}_i^T \mathbf{V} = b_i, i = 1, \ldots, l. \end{aligned} \tag{13}$$

(13) is a convex quadratic problem with linear equality constraints, and a closed-form solution can be derived using Lagrange multipliers [Strang 1986] (see details in the supplementary).

Shape Variation Constraint. After each iteration of the alternating minimization procedure, we regularize the deformation of each panel, such that the panel shape changes only within the allowable space defined by the learned shape variation model. For each panel $\mathbf{v}_i$, we first minimize the fitting energy function w.r.t. $\boldsymbol{\beta}_i$,

$$\mathbf{F} = \tfrac{1}{2}\|\bar{\mathbf{s}}_i + \mathbf{U}_i\boldsymbol{\beta}_i - \mathbf{v}_i\|^2 + \tfrac{\psi}{2}\boldsymbol{\beta}_i^T \boldsymbol{\Lambda}_i^{-1} \boldsymbol{\beta}_i, \tag{14}$$

where $\{\bar{\mathbf{s}}_i, \mathbf{U}_i\}$ are the parameters of the shape variation model for *importance* M_{P_i}, and $\boldsymbol{\Lambda}_i \in \mathbb{R}^{3\times3}$ is a diagonal matrix containing the corresponding eigenvalues. The first term in (14) encourages the new shape to be close to the current shape, while the second term is a regularization term weighted by ψ (empirically set to 0.5). The optimal shape parameter $\hat{\boldsymbol{\beta}}_i$ is obtained by minimizing (14):

$$\hat{\boldsymbol{\beta}}_i = (\mathbf{I} + \psi\boldsymbol{\Lambda}^{-1})^{-1}\mathbf{U}_i^T(\mathbf{v}_i - \bar{\mathbf{s}}_i). \tag{15}$$

We then clamp $\hat{\boldsymbol{\beta}}_i$ using (2), and recover the new shape of the i-th panel with (1). Finally, the new $\mathbf{V}$ is obtained using the new geometries of all panels. For a vertex shared by more than one panel, its coordinates are computed as the average of all corresponding vertices. To ensure the validity of the resulting layout mesh, we further impose the boundary and collinearity constraints by directly manipulating $\mathbf{V}$, as described earlier.

Interactive Refinement. The computational efficiency of our layout optimization enables us to support interactive refinement of the final layout. In particular, the user can edit specific artworks by translating or scaling them in the resulting layout. With the manipulated artworks fixed, our approach will optimize the remaining artworks and layout mesh, and immediately present the updated result to the user.

4.4 Final Layout Generation

The final layout is rendered from $\{\mathcal{T}, \mathbf{V}\}$ as follows. To create white space separating adjacent panels, each panel geometry is shrunk inward, by translating each edge along its normal direction by a certain distance d. In our implementation, we set the horizontal and vertical spacing to 10 and 5, respectively. Next, each artwork is transformed using $\mathcal{T}_i$, and clipped against its corresponding panel. Since our optimizer maximizes the fitness between the image geometry and corresponding panel geometry, this clipping should keep important regions of each artwork within the panel, while discarding trivial regions that fall outside.

Fourth Wall Break. The “fourth wall break” effect, where a character breaks the boundaries of its panel and leaks into the neighboring space, is often exploited by professional artists to augment the impact of the contents [McCloud 2006]. To achieve this effect, we extract the foreground object using an interactive image segmentation technique [Chen et al. 2012], which takes as input a box around the foreground object. We use the same ROI selected by the user in the constrained UI as the box input. The extracted foreground object is then composed into the final layout. The rightmost layout in Figure 1 and the top row in Figure 10 demonstrate this effect.

Generating Multiple Suggestions. Inspired by [Merrell et al. 2011], multiple optimized layouts are presented to the user, allowing them to easily explore different layout configurations. To enable this functionality, we rank all the initial layout samples by their MAP scores, and remove duplicates by comparing the splitting configurations, resulting in a diversified list of candidate layouts. The layout optimization is applied to the highest-ranking layouts, which are then presented to the user for interactive refinement.

5 Results and Evaluation

We have implemented our manga layout algorithm as a software tool, and run it on a PC with an Intel i7 3.1GHz CPU and 6GB RAM. On average, the initial layouts are generated in $\sim$3 seconds, and our alternating optimizer converges in $\sim$1 second for a single layout. The top-5 ranked initial layouts are further optimized to form the suggestions presented in our tool. The reading order of all the layouts presented in this paper is from left to right and then top to bottom, unless noted otherwise.

5.1 Layout Results for "*Fairy Tail*" Manga Series

We first present the layout results when the style models are trained using the "*Fairy tail*" manga series, which is representative of recently published manga. Figure 1 shows several layouts generated by our tool using the "*Fairy tail*" style. The resulting layouts faithfully respect the user-specified semantics of each artwork. More important artworks are properly magnified, while less important ones are smaller. The clear size contrast between panels of different importances directs the reader's attention to semantically important artworks. Semantically related panels identified by the user (e.g., the rightmost example) are arranged along a straight line and occupy the entire row, thereby easing the reader's effort to mentally regard them as a whole. Moreover, varying layout structures as well as irregular panel shapes further improve the visual appeal of the layouts. Finally, interesting contents in each artwork are also well preserved. These layouts illustrate that our approach can properly model and balance the important features in manga layout.

5.2 Layout Results for Other Manga Series

To demonstrate the generality of our approach, we have also trained the style models on another manga series, "*Detective Conan*". The top row of Figure 10 presents two examples with the style of "*Detective Conan*". In contrast to "*Fairy Tail*", the "*Detective Conan*" layouts more often exploits a local layout pattern consisting of three panels spanning one row (e.g., the last three panels in the right example), where one large panel in the first column is followed by two smaller panels in the second column. Furthermore, we compared the synthesized results with existing "*Detective Conan*" manga pages. The comparisons are presented in Figure 11. They show that, given a reasonable semantic specification, our approach can synthesize layouts that are functionally and visually similar to existing ones. Finally, Our approach can also handle the style of traditional Western comics (see bottom row of Figure 10).

5.3 User Study

To objectively evaluate the effectiveness of our manga layout algorithm, we have conducted a user study with 10 participants recruited from the computer science department at a local university. All participants occasionally read manga, but do not have any experience in creating manga. The input artworks are taken from four popular movie trailers: "*Twilight: New Moon*", "*Avatar*", "*Madagascar*", and "*Iron Man 2*". For each trailer, we manually extracted semantically important keyframes, and sequentially arranged the frames to constitute meaningful events. An average of 19 frames were selected from each movie trailer. For each trailer, the artworks were partitioned into several contiguous sets with sizes varying from 5 to 8 frames, forming an average of 3 sets per trailer. The artworks in one set will be arranged on one page. Finally, we assigned each artwork a score from 1 to 3, indicating its importance. For our study, we have trained our approach on "*Fairy Tail*".

Procedure. Each participant was asked to do the layouts in four sessions, one for each trailer. In each session, the input artworks were arranged by either a manual tool, which is similar to the commercially-available MangaStudio, or our automatic tool. Using the manual tool, the participants needed to first manually split the page into a desired number of the panels by drawing separating lines, and then manipulate each artwork to place the interesting contents into its panel. Our tool is employed every other session, and is used for the first session by every other user. To ensure a fair evaluation, we disabled our "fourth wall break" effect because the manual tool did not naturally support such functionality. Before the study, all the participants were given a short tutorial on each tool

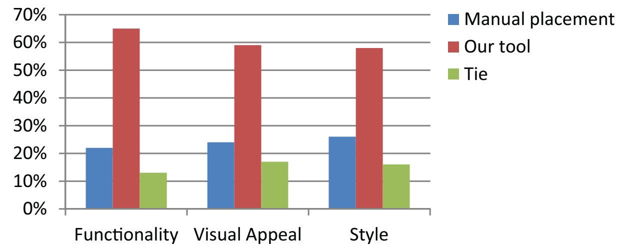

Figure 12: *Overall evaluation results. The vertical axis is the percentage of the votes aggregated across four sessions. The data suggest that the evaluators consistently prefer the results produced by our tool in terms of the three aspects. For all three aspects, such preference is statistically significant ($p < 0.05$), in accordance with chi-squared test (2 df, $\alpha = 0.05$).*

and a five-minute warm-up exercise using both tools. We also instructed the participants to allocate larger panels for more important artworks when using the manual tool. In total, 40 sets of layouts were generated by the participants. For each session, 5 sets were produced via the manual tool and another 5 sets were produced by our tool.

Timing and Usability. On average, the participants spent 1.36 minutes per layout using our automatic tool. In contrast, the participants were 3 times slower using the manual tool, with an average of 4.09 minutes per layout. During the experiment, we noticed that, when using the manual tool, all participants had a difficult time to determine a proper layout structure; they relied on a trial-and-error loop to achieve the desired layout, which was quite time-consuming. In contrast, our tool only required the users to specify simple semantic information, and the layout results could be easily tweaked with real-time feedback. The simplicity of our interface and quality of the resulting layouts greatly reduced the time and effort to create the manga layouts.

Evaluation. The layouts generated were evaluated through pairwise comparison by a *different* set of 10 participants who usually read manga and did not participate in the layout task. For each session, layout sets created by the manual tool were paired with those created by our tool, resulting in 100 pairs (4×5×5). The pairs were evenly distributed among the evaluators. For each pair, the two sets of layouts were displayed side-by-side on a computer screen in random order, and the evaluators were asked to assess the quality of the layouts in terms of the following three aspects:

- *functionality* – whether the layouts respect the semantics of the input artworks, deliver semantically important contents, and clearly convey understandable events.
- *visual appeal* – whether the layouts are visually pleasing.
- *style* – whether the overall impression of the layouts are similar to layouts in manga books familiar to the participants.

The evaluators indicated their preference by choosing among three options: "left is better", "the right is better", "both are the same". The evaluation results are shown in Figure 12. Perceptual similarity to artist-made manga layouts (i.e., style) can be quite subjective, and is influenced by personal tastes and experiences. However, it is interesting to note that the layouts by our tool are rated significantly higher in all three aspects evaluated. This positively confirms that our approach can reproduce the styles of professional manga layout, which is recognized and appreciated by the evaluators.

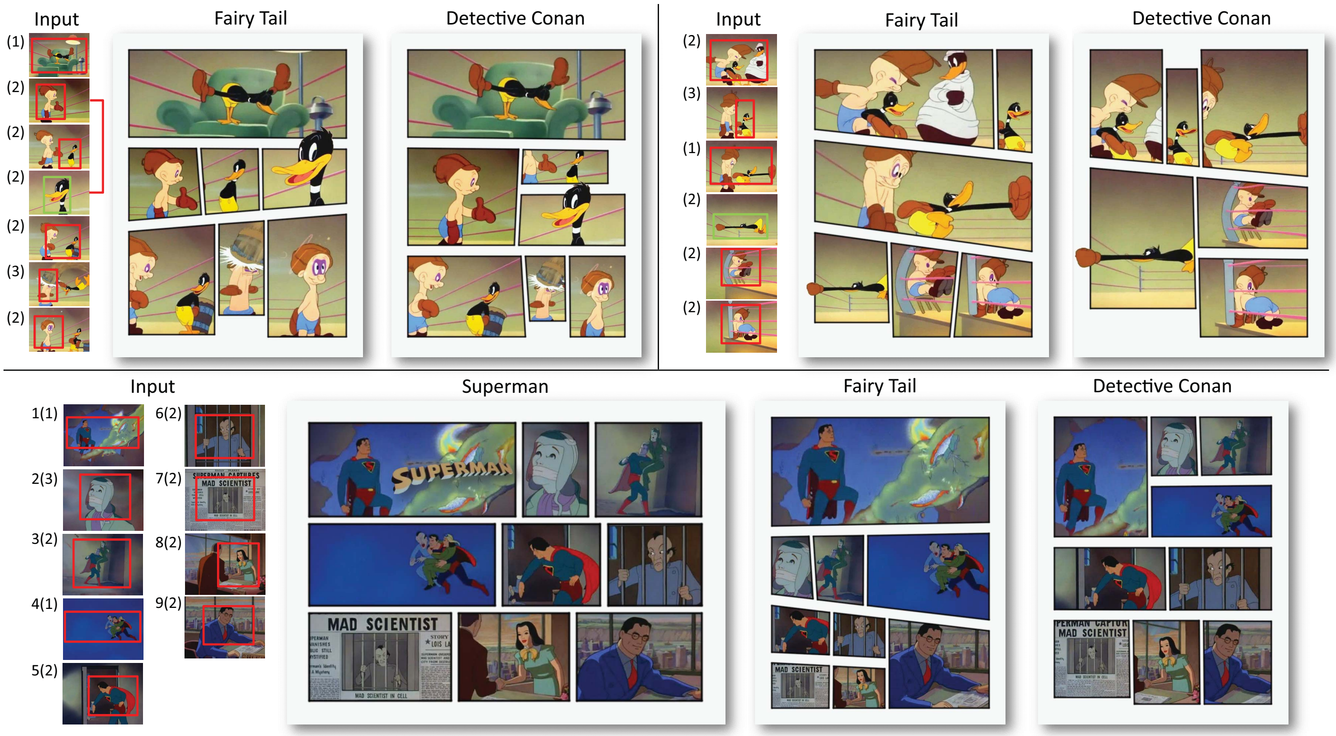

Figure 10: *The layouts generated by our approach based on styles learned from two manga series, "Fairy Tail" and "Detective Conan", and a comic strip, "Superman". Top row: input artworks ("Daffy: To Duck or Not to Duck"(1943) in the public domain) are arranged using two different manga styles. Bottom row: input artworks ("Superman"(1942) in the public domain) are arranged using one comic style and two manga styles. Note that in the bottom row, despite the regular panel shapes, the layout with the "Detective Conan" style exhibits a more complex layout pattern with the first panel spanning two rows, which contrasts with the layout using the "Superman" style.*

6 Conclusion and Discussion

We have presented a data-rich approach for automatic generation of stylistic manga layouts. We propose several models for representing the unique styles of manga layout, which can be learned from existing manga pages. Integrating the learned style models with a generative probabilistic framework and an energy optimization scheme enables our approach to rapidly produce professional-looking manga layouts that are both functionally correct and visually pleasing. This is confirmed by our user study. Generating the layouts with various styles learned from different manga series also demonstrates the generality of our approach.

Our approach is subject to several limitations. First, since this work is the first attempt to computationally reproduce layout styles of manga, we acknowledge that the proposed style models may not be complete enough to handle the styles of all types of manga, especially those with extremely expressive layouts. For example, shoujo manga contains nested panels and other layout structures, which cannot be represented by recursive splitting. Future work will explore more style models and incorporate them into our framework, which would increase the expressiveness and generality of our approach. Second, our approach currently relies on the user to decide the number of the panels per page. This may be cumbersome and labor-intensive, especially when there are hundreds of panels to be processed. How to automatically partition a sequence of artworks into pages according to high-level user specifications, such as story pacing, is another interesting and challenging problem that requires further exploration. Furthermore, it would be interesting to investigate how the pacing of the story influences layout design within individual pages or across several pages.

Finally, we envision our technique as a building block in a complete system that is able to guide novices through the essential steps of manga production, such as sketching, inking, layout, ballooning, and screening, to produce professional-looking manga. We believe that such a system would increase the accessibility of manga creation, and enable more people to enjoy the fun brought by manga.

Acknowledgments. We thank all the anonymous reviewers for their insightful comments and constructive suggestions.

References

ATKINS, C. 2004. Adaptive photo collection page layout. In *Proc. IEEE ICIP*.

BARNES, C., GOLDMAN, D. B., SHECHTMAN, E., AND FINKELSTEIN, A. 2010. Video tapestries with continuous temporal zoom. In *Proc. ACM SIGGRAPH*.

BISHOP, C. 2006. *Pattern Recognition and Machine Learning*. Springer.

CAGAN, J., SHIMADA, K., AND YIN, S. 2002. A survey of computational approaches to three-dimensional layout problems. *Computer-Aided Design 34*, 8, 597–611.

CALIC, J., GIBSON, D., AND CAMPBELL, N. 2007. Efficient layout of comic-like video summaries. *IEEE Trans. on Circuit and System for Video Technology 17*, 931–936.

CHEN, Y., CHAN, A., AND WANG, G. 2012. Adaptive figure-ground classification. In *Proc. IEEE CVPR*.

COOTES, T., TAYLOR, C., COOPER, D., AND GRAHAM, J. 1995. Active shape models-their training and application. *Computer Vision and Image Understanding 12*, 1, 38–59.

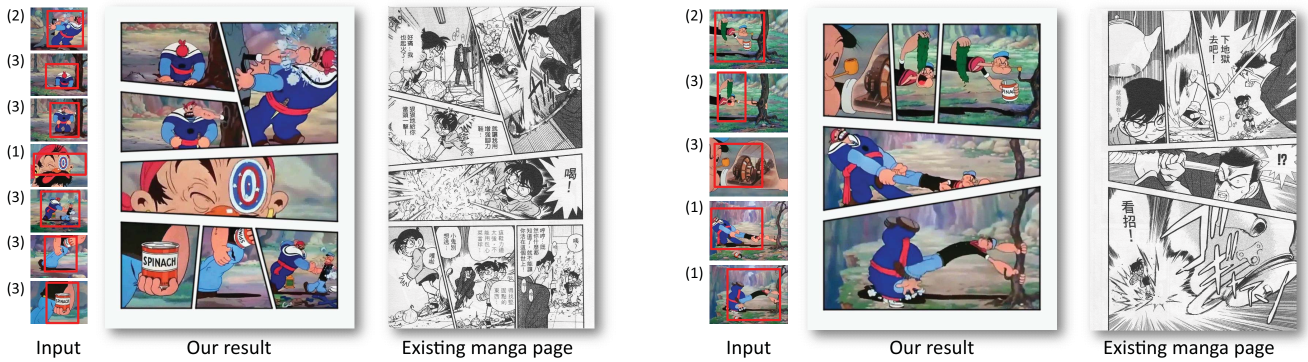

Figure 11: *Comparison with existing manga pages. Input artworks are from "Popeye: the Sailor meets Sindbad the Sailor"(1936) in the public domain. Existing manga pages are from volume 6 of "Detective Conan" manga series (©AYOYAMA Gosho / Shogakukan Inc.). Reading order here is from RIGHT to LEFT and then top to bottom.*

DURRANT, A., ROWLAND, D., KIRK, D. S., BENFORD, S., FISCHER, J. E., AND MCAULEY, D. 2011. Automics: souvenir generating photoware for theme parks. In *Proc. CHI*.

GEIGEL, J., AND LOUI, A. 2003. Using genetic algorithms for album page layouts. *IEEE Trans. on Multimedia 10*, 4, 16–27.

GEMAN, S., AND GEMAN, D. 1984. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence 6*, 6, 721–741.

GIRGENSOHN, A. 2003. A fast layout algorithm for visual video summaries. In *Proc. IEEE ICME*.

GOLDMAN, D., CURLESS, B., SALESIN, D., AND SEITZ, S. 2006. Schematic storyboarding for video visualization and editing. In *Proc. ACM SIGGRAPH*.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence 20*, 11, 1254–1259.

JACOBS, C., LI, W., SCHRIER, E., BARGERON, D., AND SALESIN, D. 2003. Adaptive grid-based document layout. In *Proc. ACM SIGGRAPH*.

KURLANDER, D., SKELLY, T., AND SALESIN, D. H. 1996. Comic chat. In *Proc. ACM SIGGRAPH*.

LOK, S., AND FEINER, S. 2001. A survey of automated layout techniques for information presentations. In *Proc. SmartGraphics*.

MANGASTUDIO, 2011. http://manga.smithmicro.com/.

MCCLOUD, S. 1994. *Understanding Comics: The Invisible Art*. William Morrow Paperbacks.

MCCLOUD, S. 2006. *Making Comics: Storytelling Secrets of Comics, Manga and Graphic Novels*. William Morrow.

MERRELL, P., SCHKUFZA, E., AND KOLTUN, V. 2010. Computer-generated residential building layouts. *ACM Trans. on Graphics 29*, 6.

MERRELL, P., SCHKUFZA, E., LI, Z., AGRAWALA, M., AND KOLTUN, V. 2011. Interactive furniture layout using interior design guidelines. In *Proc. ACM SIGGRAPH*.

NARYANAN, A. 1991. Maximum likelihood estimation of the parameters of the Dirichlet distribution. *Applied Statistics 40*, 365–374.

PREU, J., AND LOVISCACH, J. 2007. From movie to comic, informed by the screenplay. In *ACM SIGGRAPH Posters*.

PURVIS, L., HARRINGTON, S., O'SULLIVAN, B., AND FREUDER, E. 2003. Creating personalized documents: an optimization approach. In *Proc. ACM Symp. on Document Engineering*.

QU, Y., PANG, W., WONG, T., AND HENG, P. 2006. Richness-preserving manga screening. In *Proc. ACM SIGGRAPH*.

QU, Y., WONG, T., AND HENG, P. 2008. Manga colorization. In *Proc. ACM SIGGRAPH Asia*.

RIVKAH, 2006. Paneling, pacing, and layout in comics and manga #1. http://lilrivkah.livejournal.com/168859.html.

ROTHER, C., BORDEAUX, L., HAMADI, Y., AND BLAKE, A. 2006. Autocollage. In *Proc. ACM SIGGRAPH*.

RYU, D., PARK, S., LEE, J., LEE, D., AND CHO, H. 2008. Cinetoon: A semi-automated system for rendering black/white comic books from video streams. In *Proc. CIT Workshops*.

SHAMIR, A., RUBINSTEIN, M., AND LEVINBOIM, T. 2006. Generating comics from 3d interactive computer graphics. *Computer Graphics and Applications 26*, 3, 53–61.

STRANG, G. 1986. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press.

TANIGUCHI, Y., AKUTSU, A., AND TONOMURA, Y. 1997. Panoramaexcerpts: Extracting and packing panoramas for video browsing. In *Proc. ACM Multimedia*.

THAWONMAS, R., AND SHUDA, T. 2008. Comic layout for automatic comic generation from game log. In *Proc. IFIP ECS*.

TOBITA, H. 2010. Comic engine: interactive system for creating and browsing comic books with attention cuing. In *Proc. International Conference on Advanced Visual Interfaces*.

TSAI, J., 2002. Drawing manga - page layout. http://www.elfwood.com/farp/manga/manga.html.

UCHIHASHI, S., AND FOOTE, J. 1999. Summarizing video using a shot importance measure and a frame-packing algorithm. In *Proc. ICASSP*.

VIOLA, P., AND JONES, M. 2004. Robust real-time face detection. *International Journal on Computer Vision 57*, 2, 137–154.

YU, L., YEUNG, S., TANG, C., TERZOPOULOS, D., CHAN, T., AND OSHER, S. 2011. Make it home: Automatic optimization of furniture arrangement. In *Proc. ACM SIGGRAPH*.