This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

PhotoRecomposer: Interactive Photo Recomposition by Cropping

Yuan Liang, Xiting Wang, Song-Hai Zhang, Shi-Min Hu, Shixia Liu

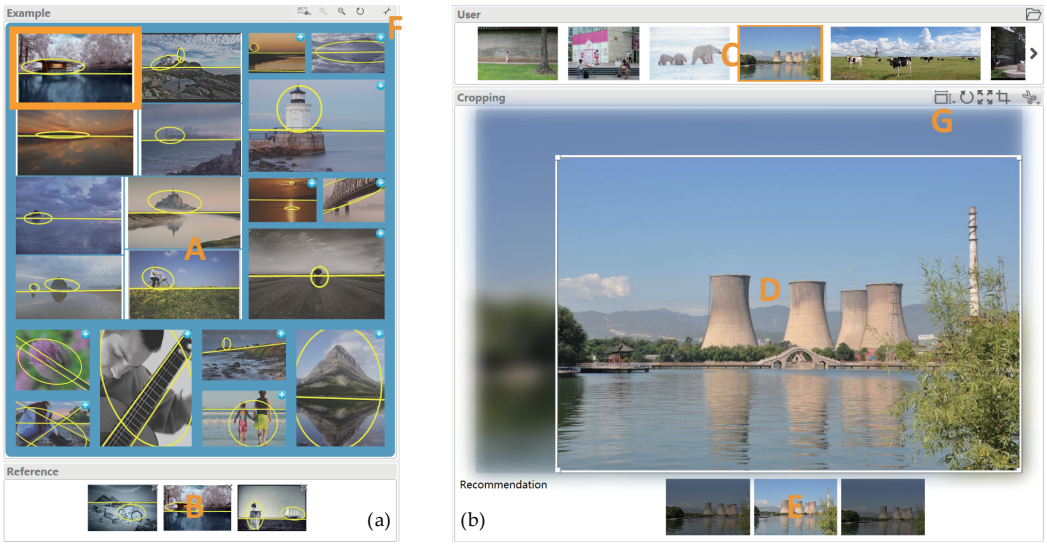


Fig. 1: Interactively recomposing photos based on examples: (a) an example view that visualizes the example hierarchy; (b) a user view that supports single/batch cropping.

Abstract—We present a visual analysis method for interactively recomposing a large number of photos based on example photos with high-quality composition. The recomposition method is formulated as a matching problem between photos. The key to this formulation is a new metric for accurately measuring the composition distance between photos. We have also developed an earth-mover-distance-based online metric learning algorithm to support the interactive adjustment of the composition distance based on user preferences. To better convey the compositions of a large number of example photos, we have developed a multi-level, example photo layout method to balance multiple factors such as compactness, aspect ratio, composition distance, stability, and overlaps. By introducing an EulerSmooth-based straightening method, the composition of each photos is clearly displayed. The effectiveness and usefulness of the method has been demonstrated by the experimental results, user study, and case studies.

Index Terms—photo recomposition, example-based learning, earth mover's distance, metric learning, photo summarization.

1 Introduction

PHOTO cropping is of significant practical use in adjusting the composition of photos due to its intuitiveness and ease-of-use [55]. In the field of photography, composition refers to the spatial arrangement of visual elements in the photo [44]. To get a better composition of a photo, many photographers employ the manual or automatic cropping function provided by commercial or free software such as Photoshop [61], Lightroom [60], or Picasa [62]. Automatic cropping typically employs a simple rule such as the rule of thirds [29], which may not always provide a desirable composition for photos. Although manual cropping can generate desired composition, it relies on user expertise and may take a considerable amount of time. Moreover, an amateur photographer may not know how to crop a photo to obtain a visually pleasing composition.

- *Yuan Liang, Xiting Wang, Song-Hai Zhang, Shi-Min Hu, and Shixia Liu are with Tsinghua University, Beijing, China. E-mail: {liangyua14,wang-xt11}@mails.tsinghua.edu.cn, {shz,shimin,shixia}@tsinghua.edu.cn.*

To solve this problem, researchers have developed a number of automatic cropping-based recomposition methods. These methods have achieved a certain level of success in improving and refining photo composition. However, without user interaction as input, they were unable to take user preferences into consideration, which is an important characteristic of practical photography [13], [38]. Interactive methods have been proposed to solve this issue, in which user interactions are utilized to capture their preferences regarding which visual elements are to be retained [54] or the placement of a certain visual element [5]. However, these methods either require an extra input device such as an eye tracker [39], or they consider photos with only a single isolated object [5]. Both drawbacks limit the practical application of these methods.

Previous studies have shown that for novices, example-based learning is more effective at reaching the desired outcomes with less investment of time and effort during acquisition [46]. On the other hand, interactive visualization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

has shed light on how to best capture user preferences [50]. Although these two techniques have shown success individually, methods that tightly integrate them for interactive, example-based photo recomposition have yet to be explored.

Two major challenges in interactive, example-based recomposition are matching photos based on their composition and summarizing a photo collection for finding the composition of interest. The key idea behind matching a photo to the example photos (with high-quality compositions) is to measure the distances between the compositions of different photos, which is difficult for two reasons. First, the positions and shapes of the visual elements (e.g., salient regions) in different photos often vary a great deal. Second, different users may have different opinions as to which visual elements are important. As a result, we need to accurately estimate the distance between different compositions based on user feedback. The example library typically contains thousands of example photos with varying compositions and aspect ratios. As a result, an effective photo summarization method that balances multiple factors is needed. Moreover, the overall composition of each photo should be displayed clearly to facilitate visual search and comparison. However, existing photo summarization methods either ignore aspect ratio [4], [43] or fail to clearly display the compositions of photos [8].

To overcome these challenges, we have developed a prototype called PhotoRecomposer to interactively recompose photos. In this paper, we focus on: 1) the distance metric used for photo cropping with example-based learning. A new metric is introduced to effectively measure the composition distance. Specifically, the earth mover's distance [48] is employed to calculate the distance between compositions. We have also developed an online metric learning algorithm to allow users to interactively refine composition distance based on their preferences; and 2) interactive photo summarization techniques that help users efficiently find an example photo representing their preferences. To effectively summarize example photos and their compositions, we have developed a multi-level example photo layout method that simultaneously considers multiple factors such as compactness, aspect ratio, composition distance, stability, and overlaps. An EulerSmooth-based straightening algorithm is introduced to clearly display the composition of each photo. Experimental results show that compared with the state-of-the-art method, our layout method is more space-efficient and more stable during exploration.

Our work makes the following technical contributions:

- **A requirement analysis** based on questionnaires and expert interviews that analyzes user requirements on photo recomposition tools.
- **An example-based recomposition method** that effectively matches user photos to example photos with earth-mover-distance-based metric learning, which aims to address user preferences.
- **An example photo layout algorithm** that effectively summarizes example photos and their compositions, with multiple design requirements for interactive exploration considered.

We first introduce related work on photo recomposition and image summarization in Section 2. Section 3 shows the results of the requirement analysis of our system, and the general system pipeline. Our proposed algorithms for recomposition and visualization modules are introduced in Sections 4 and 5. Section 6 gives a detailed evaluation of our system. Sections 7 and 8 include further discussion and the conclusion of our paper.

2 Related Work

2.1 Photo Recomposition

Cropping is widely used in photo recomposition [6]. A part of the photo is selected by leveraging the aesthetic criteria for the composition. Existing methods can be categorized into two classes: automatic methods and interactive (semi-automatic) methods.

Automatic methods recompose photos by leveraging aesthetic criteria based on the distribution of composition descriptors. In these methods, composition descriptors are first extracted with visual elements including foreground objects, prominent lines [6], attention maps [11], and over-segmentation patches [33]. Rule-based or learning-based criteria are then applied for recomposition. Commonly used rule-based criteria include empirical rules, such as the rule of thirds, the golden mean, the golden triangle, diagonal dominance [29], and symmetry [11]. Learning-based criteria model the distribution of composition descriptors with a regression model [6], [19], [51], a generative model [33], or a graph-based model [55]. Although these methods have achieved a certain level of success, modeling the diversity of real-world photography is still a challenge [19], [29]. Also, user preferences are not taken into consideration in these methods [51].

To compensate for these issues, human-computer interactions have been introduced into photo recomposition. Element-level interactions [5] are used to adjust the misextracted visual elements and capture user preferences on the placement of a certain visual element. However, the element-level interactions fail to consider multiple visual elements as a whole. Modern devices, such as eye trackers, facilitate the capture of user intents regarding which elements are to be retained in the cropping [39], [56], but the usage of such systems is limited by the device requirements.

The main difference between our work and existing interactive methods is that we model user preferences with user selection of a reference photo in an example library, which contains well-composed photos of high diversity. Given a target photo to be recomposed and a library of example photos, our method first matches the target photo to several example photos whose compositions are similar to that of the target photo. We then crop the target photo based on the matched photos. Since different users may have different aesthetic criteria, we allow users to interactively choose the matched photos and manually refine the cropping results. Moreover, to reduce user efforts, we provide a batch cropping function that propagates the cropping results of one photo to a set of photos with similar compositions. With the proposed interactive method, we modeled both the diversity of photos and user preferences with a few simple and easy interactions.

The photo recomposition problem is similar to the image thumbnailing problem [25], [42], which requires methods that automatically crop images to a fixed small size. Although these techniques generate representative thumbnails that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

are substantially more recognizable in a visual search, they cannot be directly used to optimize image compositions due to the lack of the composition analysis and recomposition methods.

2.2 Visual Summarization of Image Collections

Image summarization methods [4], [15], [26], [28], [30], [41] visually summarize images to facilitate browsing and searching. These methods can be categorized into three classes: similarity-based, compactness-based, and hybrid methods. Similarity-based methods place similar images together by using algorithms such as Principle Component Analysis (PCA) [32], Multi-Dimensional Scaling (MDS) [31], [52], or similarity trees [16], [34]. These methods effectively convey global image distribution but they are not space efficient. Compactness-based methods [45], [49] generate a space-efficient collage by avoiding blank areas and overlapping regions among the images. However, similar images are not placed close to each other. As a result, it takes users more time to locate target images [36].

Hybrid methods [4], [21] combine the advantages of similarity-based methods and compactness-based methods. Since most existing hybrid methods do not take the aspect ratio into account [4], [21], [43], they suffer from distortion or wasted screen space when the aspect ratios of images vary within dataset [8]. To solve this problem, Brivio *et al.* have developed a method based on centroidal anisotropic Voronoi diagrams [8]. This method compactly lays out images with non-uniform aspect ratios. However, this method only displays an irregular-shaped part of the image with the shape constraint of a Voronoi cell. As a result, it is difficult for users to clearly see the overall composition of images. Han *et al.* [23] proposed a tree-based layout that preserves both the aspect ratio and image shape. However, it does not ensure stability in drilling in/out operations, which makes it difficult to preserve a user's mental map during exploration. To solve these problems and effectively display images that are hierarchically organized, we have developed a multi-level image layout method, which simultaneously considers multiple factors such as compactness, the image aspect ratio, composition distance, stability, and overlaps. It also displays the overall composition clearly without occlusion by employing a EulerSmooth-based straightening technique. In addition, our method allows users to effectively recompose their images based on examples.

3 PhotoRecomposer

3.1 Questionnaire on Recomposition

We conducted a questionnaire to investigate the recomposition practices and functions required by photographers to better recompose photos. The questionnaire was distributed to four WeChat groups of photography enthusiasts in June 2016. WeChat is a cross-platform instant messaging service developed by Tencent in China. We received responses from 119 practitioners, of which 81 (68%) were male and 38 (32%) were female. Among the participants, 12 (10%) reported themselves as experts, 23 (19%) as advanced, 52 (44%) as intermediate, and 32 (27%) as novice users.

Current practices. We first asked participants how many photos they usually take and whether they crop them to refine the compositions. 61% participants reported that they usually take more than 100 photos at one time and 87% participants said they crop the photos. This indicates that many photographers can benefit from a tool that provides an effective photo cropping function.

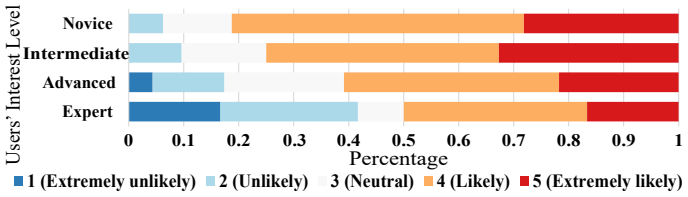


Fig. 2: Photographers' interest on the proposed tool tends to decrease as their self-reported expertise levels increase.

According to the participants, the most commonly used cropping tools are Photoshop [61] (49%), Lightroom [60] (18%), iPhoto [59] (9%), Picasa [62] (5%). Most of these tools allow users to manually crop photos by moving, dragging and rotating a cropping box. Users are provided with visual guidance such as a grid, to follow the rule of thirds [61]. While manual cropping can generate photos with a desired composition, it is mentally demanding and time-consuming. On average, the participants claimed they spend 2.9 minutes to crop each photo. Some tools such as Picasa [62] provide an automatic cropping function. However, such tools usually operate according to a simple rule, such as the rule of thirds [29]. As a result, the cropping results are not always desirable. Moreover, 57% participants regarded cropping as a subjective process. Thus, it is impossible for such an automatic cropping method to meet the different needs of different users.

Many participants showed interest in tools that can help them effectively and efficiently crop a large number of photos. We asked them to rate their interest level based on a 1-5 Likert scale (1: not at all interested, 5: extremely interested) and 71% participants returned a rating of 4 or 5. We also found that photographer interest tended to decrease as their self-reported expertise levels increased (Fig. 2).

Tool functions. To understand the requirements of photographers, we summarized functions that can potentially help them recompose photos and asked the participants to rate how important each function is (1-5 Likert scale, 1 indicates unimportant and 5 indicates very important).

Fig. 3 shows the distribution of the function ratings by the participants. The functions were categorized into two classes. The first class contains three example-library-

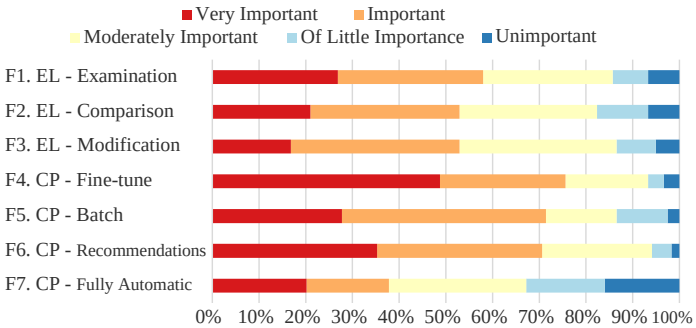


Fig. 3: Importance of different functions. Here EL means example library and CP means cropping.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

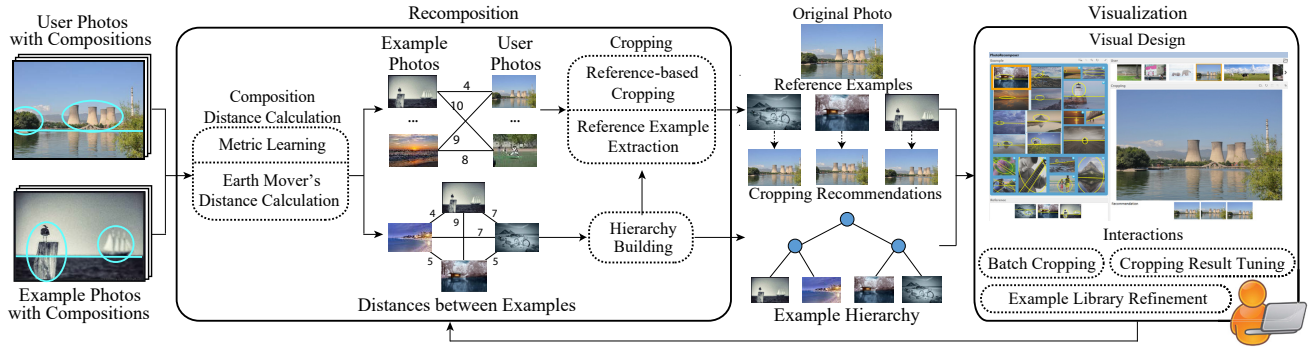


Fig. 4: An overview of PhotoRecomposer. Given a collection of user photos, example photos, and their compositions, the recomposition module generates reference examples and cropping recommendations for each photo as well as an example hierarchy. These results are fed into the visualization module, which effectively summarizes example photos and their compositions and provides rich interactions for visually exploring and recomposing the photos.

related functions. The three functions are examining example photos and their compositions (F1), comparing user photos with those in the example library (F2), and modification of the example library including adding/deleting photos and tuning the photo hierarchy (F3). More than 50% of participants consider example-library-related functions to be important or very important for them to better recompose photos. The second class consists of four functions that pertain to cropping. These functions are fine-tuning the cropping results (F4), cropping a batch of photos based on previous cropping results (F5), checking several cropping recommendations to obtain a good cropping solution (F6), and cropping photos in a fully automatic manner (F7). As shown in Fig. 3, more than 70% of participants agreed that F4, F5, and F6 were (very) important. However, fully automatic photo cropping was not well accepted among the participants. This is not surprising since a majority (57%) of participants consider cropping to be a subjective process.

3.2 Design Requirements

We distilled the design requirements based on the results of the questionnaire survey. The desired requirements can be classified into the following two categories.

The first category is related to the summarization, exploration, and refinement of the example library.

R1. Exploring example photos at different abstraction levels. The example library typically contains thousands of photos. The photographers we interviewed said that when they take photos, they tend to first follow a general composition rule and then apply a variant of the composition rule to accommodate different luminance, semantics or stress. Accordingly, it is natural to organize the photos in a hierarchy, where high-level categories help users gain an understanding of general composition rules, and low-level photo categories help users decide which variant of the composition rule to be used. Accordingly, we need to support photo exploration at different levels of granularity, with the aim of quickly identifying the composition(s) of interest (F1).

R2. Refining the example library based on user preferences. The composition distance is leveraged to measure the similarity of different compositions and thus hierarchically organize example photos. Since different users may have a different similarity degree for two given compositions, they prefer to interactively refine the example hierarchy (e.g., move a photo from one cluster to another one) based on their preferences (F3). In addition, the participants also requested the ability to add/remove example photos from the library.

The second category of design requirements is related to **user photo cropping**, including cropping recommendations, cropping result tuning, and batch cropping.

R3. Examining multiple cropping recommendations. More than 70% of survey participants considered cropping recommendations (very) important for effective and efficient recomposition (F6). They commented that better cropping recommendations usually trigger them and help them crop photos more quickly.

R4. Tuning cropping results. The cropping recommendations are not always perfect. Moreover, different users may have different requirements. Thus, the participants wanted the ability to fine-tune the cropping results based on their preferences (F4).

R5. Cropping a batch of photos based on previous cropping results. According to our questionnaire, many photographers take more than 100 photos at one time, some of which are similar in terms of composition. Thus, they expressed the need for a batch cropping function (F5).

3.3 System Overview

Based on the aforementioned design requirements, we have developed PhotoRecomposer. As shown in Fig. 4, our system consists of two major modules: recomposition and visualization. The recomposition module contains three sub-modules: composition distance calculation, cropping, and hierarchy building. Specifically, the input of PhotoRecomposer is example photos, user photos, and their compositions. Based on this information, the composition-distance-calculation sub-module measures the distances between the compositions of these photos. The cropping sub-module leverages the composition distances to extract reference examples (matched examples) for each user photo and generate the corresponding cropping recommendations (**R4**). To support efficient reference example extraction and multi-level exploration (**R2**), the hierarchy-building sub-module generates a hierarchy of example photos based on their composition distances. The cropping recommendations and example hierarchy are then fed into the visualization module, which effectively summarizes example photos and their compositions (**R1**), as well as supporting rich interactions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

such as example library refinement (**R3**), cropping result tuning (**R5**), and batch cropping (**R6**). After a user crops a photo or refines the example library, the cropping result or refinement is then send back to the recomposition module to learn more accurate composition distances.

The user interface of PhotoRecomposer consists of two views, each of which corresponds to a category of requirements. The first view is the example view (Fig. 1(a)), which visualizes the example hierarchy (Fig. 1A) and the reference examples (Fig. 1B). We represent each cluster in the hierarchy with an example photo and the corresponding composition (blue lines and ellipses). The depth of the hierarchy is encoded by its border thickness, where thinner borders represent deeper clusters. This view supports multi-granularity example photo exploration (**R1**) and interactive example library refinement (**R2**). The second view is the user view (Fig. 1(b)), which displays the user photos (Fig. 1C), cropping workspace (Fig. 1D) and cropping recommendations (Fig. 1E). By tightly integrating earth mover's distance calculation and online metric learning, this view supports rich interactions, including cropping recommendation examination (**R3**), single cropping result tuning (**R4**) and efficient batch cropping (**R5**).

4 Example-based recomposition

4.1 Algorithm Overview

To support real-time interactions, we divide the recomposition algorithm into an offline process and an online process. Because building a hierarchy of example photos is time-consuming, it is done in the offline process. In the online process, given a user photo, we extract the relevant reference examples and crop the user photo based on them. Accordingly, our algorithm contains the following three steps:

Hierarchy building. To efficiently handle a large example library, we first build an example hierarchy based on the composition distance between example photos (**R1**). K-Mediods clustering [35] is employed to build the hierarchy, which is widely used to cluster data with the pairwise distance.

Reference example extraction. This step selects n_e ($n_e = 3$ in PhotoRecomposer) reference examples whose compositions are similar to a given user photo. To support real-time interactions, we employ beam search [20] to retrieve similar examples. Beam search is a heuristic search algorithm that can quickly return search results by only searching the subtrees that are most similar to the user photo.

Reference-based cropping. The major goal of this step is to crop the user photo I_u based on the reference examples $\mathbb{I}_e$ (**R3**, **R5**). Without loss of generality, we use the example of cropping I_u by a reference $I_e \in \mathbb{I}_e$ to illustrate the basic idea. According to discussions with several photographers and previous work [22], following criteria are considered:

- the cropping result $Crop(I_u)$ and the reference example I_e should have similar compositions;
- avoiding too small of a cropping window because it may lose some important visual elements;
- avoiding too large of a cropping window because the user is unsatisfied with the current composition.

To this end, we define an energy function $E_{\text{crop}}(.)$ that jointly considers these criteria:

$$E_{\text{crop}}(Crop(I_u)) = d(Crop(I_u), I_e)/(r^{\alpha}(1-r)^{\beta}),$$

where $d(Crop(I_u), I_e)$ is the composition distance between the cropping result $Crop(I_u)$ and I_e, r is the ratio between the sizes of the cropping result and the original user photo, and a larger α (β) prevents the cropping window from being too small (large). Parameters α and β are empirically set as 0.15 and 0.05, respectively. We employ an exhaustive search to find the best cropping result that minimizes $E_{\text{crop}}(.)$. In our scenario, the real-time response to user interactions is ensured because we only need to search for four parameters: the x and y coordinates of the top-left corner, the width, and the height of the cropping window. We follow Yan *et al.* [51] to split each parameter into 33 uniform intervals in the search space.

It can be seen from the above discussion that the composition distance between photos is the key to building the example hierarchy, extracting reference examples, and finding the best cropping result. In the next subsection, we discuss: 1) how to accurately calculate the composition distance; and 2) how to update the composition distance based on user feedback to support interactive refinement (**R2**).

4.2 Distance Formulation and Calculation

As the positions and shapes of the visual elements in different photos often vary a great deal, it is not trivial to accurately measure the composition distance between photos. To estimate the distance, we first design a consistent representation for different visual elements. We then find correspondences between the visual elements in different photos based on the earth mover's distance [48]. Finally, we use the correspondences to calculate the composition distance. Accordingly, our algorithm contains the following three steps:

Visual elements extraction and representation. We use the positions and shapes of salient regions and prominent lines to model the composition of a photo because they are the most important visual elements in a photo [6]. We employ a recent widely used method, global-contrast-based method [12], to extract salient regions. Each salient region is represented by an ellipse. A prominent line is a set of line segments that tends to visually stand out in perception. The prominent lines are extracted by combining LSD detection [47] and DBSCAN [17], which is a common practice in computer graphics [29]. In this way, the composition of I is modeled by a set of salient regions and prominent lines.

Finding correspondences between the visual elements. This step aims to find accurate correspondences between visual elements of two photos I_i and I_j. Due to occlusion and clutter in photos, we allow partial correspondence, which means a salient region (prominent line) in I_i can be matched to two or more salient regions (prominent lines) in I_j. To this end, we employ the earth mover's distance (EMD) [48] to find accurate correspondences, which naturally allows partial correspondence.

The EMD measures the distance between two sets of visual elements in the source photo (I_i) and the destination photo (I_j). Here we take salient regions as an example to illustrate the basic idea. The EMD of salient regions measures the minimum amount of cost to transform the distribution of the earth (salient regions) in the source photo into that in the destination photo:

$$C_s(I_i, I_j) = \min \sum_{k,l} M_{i\to j}^{k\to l} c_{i\to j}^{k\to l}. \quad (1)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

Subject to:

$$\forall k : \sum_{l} M_{i\to j}^{k\to l} = m_i^k, \text{ and } \forall l : \sum_{k} M_{i\to j}^{k\to l} = m_j^l. \quad (2)$$

Here, m_i^k is the area of salient region s_i^k in I_i, $M_{i\to j}^{k\to l}$ denotes the amount of earth moved from s_i^k to s_j^l, and $c_{i\to j}^{k\to l}$ defines the cost of moving one unit of earth from s_i^k to s_j^l (cost function). Supposing we already know the value of each $c_{i\to j}^{k\to l}$, this problem is a linear programming problem, which can be solved by Mosek [1]. Next, we discuss how to determine the cost function $c_{i\to j}^{k\to l}$.

It is natural to assume that the longer the distance between s_i^k and s_j^l, the more it costs to move s_i^k to s_j^l. In addition, we assume that the more difference between the shapes of s_i^k and s_j^l, the more it costs to move s_i^k to s_j^l. Based on these assumptions, we define the following cost function, which consists of five terms, each corresponding to one dimension:

$$c_{i\to j}^{k\to l} = |x_i^k - x_j^l| + |y_i^k - y_j^l| + |a_i^k - a_j^l| + |b_i^k - b_j^l| + |\theta_i^k - \theta_j^l|, \quad (3)$$

where x, y, a, b, θ are the x-coordinate, y-coordinate, the length of a major radius, the length of a minor radius, and the orientation of a salient region, respectively.

To accommodate different scales of different terms in Eq. (3), each term is then generalized. We rewrite the term corresponding to the x-coordinate of a salient region in Eq. (3) as:

$$c_x(s_i^k, s_j^l) = |x_i^k - x_j^l| = |\int_{x_j^l}^{x_i^k} 1dx|. \quad (4)$$

If we replace 1 by a function $f(x;\phi_x)$ with learnable parameters ϕ_x, we can learn its parameters based on user feedback. In this way, the term corresponding to x-coordinate of a salient region in Eq. (3) is generalized as:

$$c_x^{new}(s_i^k, s_j^l) = |\int_{x_j^l}^{x_i^k} f(x;\phi_x)dx|. \quad (5)$$

There are several ways to parameterize $f(x;\phi_x)$, such as setting $f(x;\phi_x)$ to a polynomial function or a step function. For efficiency's sake, we set $f(x;\phi_x)$ as a step function.

$$f(x;\phi_x) = \phi_{x,q}, x \in [x_{q-1}, x_q), q = 1, 2, 3 \ldots N_x, \quad (6)$$

where $x_q = qw/N_x$, w is the width of the photo, N_x is the number of intervals in the step function, and $\{\phi_{x,q}\}$ are the learnable parameters. To balance accuracy and overfitting, N_x is set to 12 with a grid search [7]. Other terms in Eq. (3) can be generalized and parameterized in the same way.

The EMD of prominent lines can be calculated in the same way. The only differences are that: 1) the cost function (Eq. 3) contains only three terms: x, y coordinates of the midpoint of the line, and its orientation θ; and 2) m_i^k is replaced by the length of prominent lines.

Composition distance calculation. Here, we use the EMD of salient regions $C_s(I_i, I_j)$ and prominent lines $C_l(I_i, I_j)$ to calculate the composition distance between different photos. We define the composition distance of two photos I_i and I_j as:

$$d(I_i, I_j) = C_s(I_i, I_j) + C_l(I_i, I_j) + d_a(I_i, I_j). \quad (7)$$

The third term $d_a(I_i, I_j)$ measures the difference between the aspect ratios of the two photos, which is parameterized the same way as Eq. (5).

4.3 Metric Learning

Next, we discuss how to learn the parameters in the generalized cost function based on user feedback.

In PhotoRecomposer, users can provide two types of feedback. First, a user can move a photo I from a photo cluster A to another cluster B. In this case, the composition distance between I and any photo in A (B) should be larger (smaller) than the current distance. Second, a user can accept or reject a cropping recommendation. In this case, the composition distance between the user photo and the example photo should be larger (smaller) if the user rejects (accepts) a cropping recommendation. Both kinds of user feedback provide a set of "must-link" constraints $\mathbb{M} = \{m_i\}$ and "cannot-link" constraints $\mathbb{C} = \{c_j\}$.

The composition distance between each "must-link" ("cannot-link") pair should be smaller (larger) than the current distance.

To learn the parameters based on user feedback, we should satisfy the constraints provided by the user and make sure the new parameters should be close to the current parameters [50]. As there may be conflicts in user feedback, we introduce slack variables as in [48]. Mathematically, we formulate it as a constrained quadratic programming problem:

$$\min \sum_i \xi_i + \sum_j \eta_j + \lambda_1 \sum ||\phi - \hat{\phi}||_2 + \lambda_2 \sum ||\phi||_2. \quad (8)$$

Subject to

$$\forall m_i \in \mathbb{M}, d(m_i) < \hat{d}(\mathbb{M}) + \xi_i, \xi_i \geq 0;$$
$$\forall c_j \in \mathbb{C}, d(c_j) > \hat{d}(\mathbb{C}) - \eta_j, \eta_j \geq 0.$$

Here $\{\xi_i\}$ and $\{\eta_j\}$ are the slack variables, ϕ is the parameter to be learned, which is defined by Eq. (6) for each specific dimension (e.g., x-coordinate of salient regions), $\hat{\phi}$ represents the previous value of ϕ, and $\hat{d}(\mathbb{M})$ ($\hat{d}(\mathbb{C})$) denotes the average distance between the "must-link" ("cannot-link") pairs before the user feedback is provided. The first and second terms aim to maximally satisfy user feedback. The third term ensures that the new parameters are close to the current ones. The last term is introduced to prevent overfitting. λ_1, λ_2 are the parameters that balance these terms. We set $\lambda_1 = \lambda_2 = 1$ using a grid search, which aims to get an optimal gain in Normalized Mutual Information (NMI) [18]. This optimization problem can be efficiently solved with the interior point method [1].

To support real-time interactions, we do not update the whole hierarchy when the parameters are updated. Instead, we only check the photos that are most relevant to clusters A and B to speedup the algorithm. Specifically, after the user moves a photo from cluster A to cluster B, the algorithm 1) checks all children of A and determines whether they should be moved to As sibling clusters (infer more cannot-link pairs related to A), and 2) checks all children of Bs siblings and determines whether they should be moved into B and then makes corresponding suggestions (infer more must-link pairs related to B). The whole pipeline of updating the parameters and suggesting nodes to be moved takes 0.6s on average. We show such an adjustment case in the supplemental material. However, since the updated parameters influence all the pairwise distances in the example library, the aforementioned adjustment only approximates the changes needed in the example library with the updated parameter.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

Such an approximation is effective in refining the example library, as shown in Sec. 6.1.1.

5 VISUALIZATION

Based on the requirements discussed in Sec. 3.2, we designed a visualization that consists of two parts. The example view (Fig. 1(a)) meets example-library-related requirements, including exploration (**R1**) and refinement (**R2**) of the example library. The key challenge here is to design a layout algorithm that effectively summarizes the example photos and their compositions. The user view (Fig. 1(b)) meets user-photo-related requirements (**R3** – **R5**) by providing rich interactions. Next, we will describe the example hierarchy layout algorithm and how interactive exploration and recomposition are supported.

5.1 Example Photo Layout

To effectively summarize example photos and help users quickly identify compositions of interest, we have formulated the layout as an optimization problem.

5.1.1 Criteria and Constraints

Our layout goal is defined by four criteria that are commonly considered in state-of-the-art image summarization methods.
Compactness [27]: Generate a compact and space-efficient layout by minimizing the empty space ratio $E_c = S_w/S$, where S_w is the white space size and S is the total area size.
Distance preservation [43]: Photos with similar compositions should be placed together to facilitate exploration [36]. Accordingly, this criterion minimizes the difference between the geometric distance in the layout and the composition distance: $E_d = (|P_i - P_j| - d_{ij})^2$. Here P_i, P_j represent the positions of photos i, j and d_{ij} is their composition distance.
Stability [14]: To preserve a user's mental map, adjacent layouts during interactions should be as stable as possible. Thus, $E_s = \sum_i |P_i - P_i'|^2$ should be minimized, where P_i' is the position of photo i in the previous layout.
Non-rectangular boundary reduction [8]: Non-rectangular boundaries of the photos (Fig. 5(b)) should be avoided to improve overall composition readability. Accordingly, we minimize $E_r = \sum_i |N_i - 4|$, where N_i is the number of points on the layout boundary of photo i and 4 is the number of edges in a rectangle.

According to design requirement **R1** and existing literature on composition and photo analysis, the layout must satisfy the following three hard constraints for better displaying compositions.
Hierarchical constraint C_h: The layout should convey the hierarchical structure of the compositions by placing the photos with the same parent together. Thus the layout should satisfy $\Omega_{\text{child}} \subset \Omega_{\text{parent}}$ for each pair of nodes $(\text{child}, \text{parent})$, where Ω_{node} is the layout region of the node.
Non-overlapping constraint C_o: If photos overlap with each other, only a portion of the photo is displayed and it is difficult for users to see the overall compositions. Thus, we need to avoid any overlap to improve photo readability [23], which means for each pair of nodes (n_1, n_2) without ancestral relationships, they need to satisfy $\Omega_{n_1} \cap \Omega_{n_2} = \emptyset$.
Aspect ratio constraint C_a: To faithfully reveal the photo composition without distortion, each photo should be shown with its original aspect ratio [8].

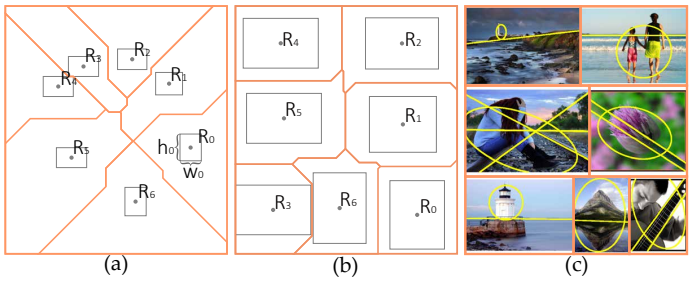


Fig. 5: Iterative steps of the photo layout algorithm: (a) L_∞-norm Voronoi tessellation with rectangular sites; (b) Approximate layout after the Voronoi-tesselation-based layout estimation step; (c) Final layout after the Euler-smooth-based boundary straightening step.

5.1.2 Constrained Optimization

By combining the above criteria and constraints, we have formulated the layout as a constrained optimization problem:

$$E = E_c + \lambda_d E_d + \lambda_s E_s + \lambda_r E_r,\ s.t., C_h, C_o, C_a. \quad (9)$$

Here $\lambda_d, \lambda_s, \lambda_r > 0$ are the parameters to balance different criteria. In our implementation, $\lambda_d, \lambda_s, \lambda_r$ are set to empirical values 1, 0.1, and 100 respectively, to balance different variances of these criteria.

The difficulty with the constrained optimization problem lies in the set of constraints that must be handled simultaneously since these constraints can compete with each other during the layout process [21]. To tackle this issue, we combine the advantages of Voronoi treemap [2] and EulerSmooth [40].

Given a display area, our approach satisfies the hierarchical constraint C_h by employing the Voronoi treemap layout. Specifically, we recursively subdivide the display area and compute the layout area for each cluster from top to bottom of the hierarchy. The constrained optimization problem is then divided into sub-problems without C_h. For each sub-problem, we reduce the search space by dividing it into two parts. First, we use a **Voronoi-treemap-based layout estimation** with a gradient-based solver, as it is highly efficient at non-linear optimization, to calculate an approximate layout area for each photo without considering the non-rectangular boundary issue (E_r, as it is discrete and non-differentiable). Second, we propose an **EulerSmooth-based boundary straightening** method to remove the non-rectangular boundaries while also minimizing E.
Voronoi-tesselation-based layout estimation. We employ Voronoi tesselation to compute an approximate layout because it is able to generate a space-efficient parameterized layout without overlapping (C_o). To control the aspect ratio (C_a) and avoid curvy boundaries for each Voronoi cell, we employ the L_∞-norm metric (Tan *et al.* [43]) in Voronoi tessellation, in which each Voronoi site is weighted by the aspect ratio measure used by Brivio *et al.* [8]. The layout is then parameterized by the position x_i, y_i and size w_i, h_i for each site. The final metric is written as:

$$d_{L_\infty}(x, y, R_i) = \max(|x - x_i| - w_i/2, |y - y_i| - h_i/2) \quad (10)$$

An optimal solution of parameters x_i, y_i, w_i, h_i can then be arrived at with the gradient projection method in nonlinear programming [37]. A detailed derivation is in the supplemental material.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

EulerSmooth-based boundary straightening. Since the approximate layout results (e.g., Fig. 5(b)) contain non-rectangular boundaries that prevent users from easily interpreting the overall compositions, we propose a boundary straightening method, which consists of the following steps:
Step 1: Control point selection. For each photo i with non-rectangular shape, we represent its boundary by using a polygon $B_1B_2...B_n$, where B_m $(1 \leq m \leq n)$ is a point on the boundary. Each B_m that satisfies $\angle B_{m-1}B_mB_{m+1} < \pi$ is then identified as a control point to be adjusted.
Step 2: Control point adjustment. Given a control point B_m, we can adjust the position of B_m to remove wiggles. As shown in Fig. 6, we first move B_m to the centroid of $\triangle B_{m-1}B_mB_{m+1}$, as dictated by EulerSmooth [40]. The positions for B_{m-1} and B_{m+1} are then updated to satisfy $B_{m-1}B_m /\!\!/ B'_{m-1}B'_m$ and $B_mB_{m+1} /\!\!/ B'_mB'_{m+1}$. This adjustment is repeated until 1) two points (e.g., B_{m-1} and B_m) have collapsed into one; or 2) the threshold of the iteration number is reached. In the first case, a wiggle is removed.

We repeat Step 2 to reduce the number of wiggles iteratively. A key here is to select the wiggle point that is to be adjusted. To minimize E, we employ a greedy method, which selects the wiggle point leading to the minimal E at each iteration. As the greedy method ensures E to decrease at each iteration, the algorithm is guaranteed to converge to a local minimum if a minimal E exists. A final layout is shown in Fig. 5(e).

However, this method only reduces the number of points in the boundaries, which implies that it does not work for a photo initially placed with a triangular boundary. The situation happens with a frequency of less than 2% in our experiments, thus a simple re-layout with a group of perturbed initial values typically solves this issue.

The above steps for generating a layout takes 1.3s on average. Fig. 7 compares our method with other image summarization methods. As shown in the figure, grid layout, squarified treemap [9], and optimal rectangle packing [24] are free from overlaps. However, there are relatively large white spaces in the layout results of these methods. Picture collage eliminates any white spaces, but there are overlaps in the layout, which hinders the readability of the composition (e.g., Fig. 7A). Our method solves these issues by jointly considering multiple criteria and constraints.

5.2 Interactive Exploration and Recomposition

The following interactions are provided to assist 1) exploration of the example library and 2) effective and efficient recomposition of user photos.
Exploration of the example library (**R1**). The example library provides an easy way for users to find inspiration for recomposition and gain understanding of good compositions. For example, a user can select one of the reference examples in Fig. 1B and set it as the focus. Our tool then shows related compositions in the example view in a focus+context manner. It allows the user to explore the example view by drilling into a cluster or clicking the drill out button (Fig. 1F) to return to high-level views. If the user finds examples that are useful for cropping, s/he can set them as the new references. Our tool then updates the cropping recommendations based on the user-selected references.

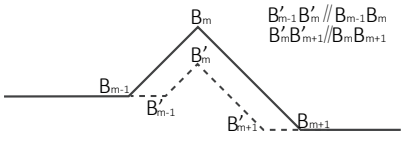


Fig. 6: Adjusting the positions of the boundary points to gradually remove wiggles. Here B_{m-1}, B_m, B_{m+1} are boundary points before adjustment, and B'_{m-1}, B'_m, B'_{m+1} are the points after adjustment.

Example library refinement (**R2**). We allow users to refine the example hierarchy by dragging and dropping clusters (or photos). After the refinement, the composition distances are updated and directed edges that imply further refinement recommendations are displayed. The user can then click the edge to browse the recommendations and make further refinements. Users can also add/remove example photos.
Single cropping (R3, R4). To crop a photo, a user can start by examining and comparing cropping recommendations (Fig. 1E). S/he can crop the photo quickly by accepting a recommendation, or explore the example view to find inspirations. We also allow users to fine-tune the cropping results by dragging the cropping box in the cropping workspace (Fig. 1D).
Batch cropping (R5). If a user takes many photos in similar scenes, our tool can help her/him quickly recompose these photos via batch cropping. Specifically, a user can perform a batch crop by selecting a previous cropping result from the history panel (Fig. 10(b)). PhotoRecomposer then finds photos with similar compositions and recomposes these photos based on the user-selected cropping result. The cropping recommendations for these photos are ranked by the recommendation certainty (Fig. 10(a)). The user can browse the ranked list and quickly determine which cropping recommendations to accept.

6 Evaluation

In this section, we first evaluate the metric learning algorithm and example photo layout algorithm by conducting two quantitative experiments. Next, the effectiveness of our tool is demonstrated through a user study with 16 photographers. Finally, typical cases found by photographers during the user study are introduced to show the usefulness and effectiveness of PhotoRecomposer. The following datasets were used in the evaluation:

- **Dataset A** contains 2,753 example photos. These examples were manually collected from two websites: The Canadian online photography community and marketplace 500px [57], and digital photography challenge website DPChallenge [58]. We iteratively divided these photos into K clusters by using K-mediods and generated a five-level hierarchy with 367 internal nodes. Here, K is determined by using the Elbow method [10].
- **Dataset B** contains 273 photos that have not been cropped. These photos belong to miscellaneous categories (e.g., architecture, nature, and people) and were collected from the user study participants.
- **Dataset C** contains 43 photos that were taken during college baseball games. A user study participant provided this dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

(a) (b) (c) (d) (e)

Fig. 7: Comparison of different methods to layout compositions: (a) grid layout, (b) squarified treemap [9], (c) optimal rectangle packing [24], (d) picture collage [49] and (e) our layout. Our layout achieves the high compactness without overlapping.

All the experiments were conducted on a workstation with an Intel Xeon E5630 CPU (2.53 GHz) and 32GB of memory.

6.1 Quantitative Experiments

6.1.1 Metric Learning

This experiment aims to evaluate the effectiveness of the metric learning algorithm at refining the example library.

Experimental settings. To evaluate how the algorithm performs on different types of photos, we selected three smaller subsets (A_1, A_2, A_3) of example photos from dataset A, as getting the ground-truth of the whole dataset with agreement is a time-consuming task. A_1 contained only scenery photos ($N = 133$, $D = 2$), A_2 contained only human portrait photos ($N = 244$, $D = 3$), and A_3 contained various kinds of photos, including scenery, human portraits, and pet photography ($N = 167$, $D = 3$). Here N denotes the number of photos in the dataset, and D is the depth of the ground-truth hierarchy. We recruited two professional photographers from a college photography association to manually label the ground-truth hierarchies for A_1, A_2, and A_3. For each layer, the two photographers first discussed the general principle for classification, then manually labeled the photos individually, and finally solved the conflicts via discussion.

Criteria. To measure the effectiveness of our metric learning algorithm, we adopted two metrics. The first was the **number of refinement steps (N_r)** needed to change the initial example hierarchy to the desired one. In each refinement step, we randomly selected a photo that was incorrectly clustered and moved it to the correct cluster. Next, we used the metric learning algorithm to recalculate the composition distances and updated the example hierarchy. This process was repeated until the desired example hierarchy was generated. Here the desired example hierarchy is the ground-truth hierarchy labeled by the professional photographers. A smaller N_r indicates better performance of the algorithm. To reduce bias caused by the refinement order, we repeated the experiment 100 times with different refinement orders. The second criterion was the **number of photos correctly moved (N_c)** at each refinement step, where a movement is "correct" when the move increases NMI. A larger N_c suggests that the algorithm was more effective.

Results. Table 1 compares our metric learning algorithm (ML) with a baseline that does not use metric learning (NoML). The results show that our algorithm outperformed the baseline on all datasets. Note that for the baseline, without metric learning, the system does not suggest any other photo be moved when the user moves a photo. Thus at each step only one photo is moved to a correct cluster (the one manually moved by the user), which indicates the value of N_c is always 1.0. On average, our algorithm was able to reduce 51% user effort compared with the baseline. User effort is measured by the number of refinement steps. This demonstrates that our metric learning algorithm is able to learn more accurate compositions based on user feedback. All the results produced by ML have reasonable standard deviations. It indicates our algorithm has decent results regardless of the random order of user operations.

TABLE 1: Comparison of our metric learning algorithm (ML) and the baseline (NoML) on three datasets.

	Dataset A_1		Dataset A_2		Dataset A_3	
	N_r	N_c	N_r	N_c	N_r	N_c
NoML	53.0 ± 0	1.0 ± 0	147.0 ± 0	1.0 ± 0	77.0 ± 0	1.0 ± 0
ML	$\mathbf{28.9 \pm 3.0}$	$\mathbf{3.0 \pm 2.1}$	$\mathbf{62.6 \pm 5.0}$	$\mathbf{3.3 \pm 2.0}$	$\mathbf{45.5 \pm 5.7}$	$\mathbf{1.7 \pm 1.2}$

6.1.2 Example Photo Layout

In this experiment, we demonstrated the effectiveness of the example photo layout algorithm by comparing it with four baseline methods.

Experimental settings. We selected four widely used image summarization methods as the baseline methods: grid layout (GL), squarified treemap [9] (ST), rectangle packing [24] (RP), and picture collage [49] (PC). The example layout results of these methods are shown in Fig. 7.

We compared our algorithm with these methods over the course of 500 trials. Each trial mimicked a case from the layout of PhotoRecomposer. In each trial, a photo was first randomly selected from the hierarchy of Dataset A as the focus of the layout. Then the evaluation algorithm performed a random interaction, including drilling in, drilling out, and moving a photo in the hierarchy to evaluate the stability of the layout.

Criteria. The layout quality was measured by using 7 criteria. The first three criteria were **compactness** (E_c), **distance preservation** (E_d), and **stability** (E_s), which are detailed in Sec. 5.1.1. The fourth criterion was **photo overlapping ratio** $E_o = S_o/S$, where S_o denotes the overlapping area size and S is the total area size. The fifth and sixth criteria were neighbourhood preservation (E_p) and orthogonal alignment (E_a) [21]. We set the number of neighbours as 25% of the number of photos for both criteria. We invited 7 users to browse the hierarchies of example photos with the layout.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

TABLE 2: Comparison of our layout algorithm and the baselines. E_c, E_d, E_s, E_o, E_p, E_a, and P measure layout quality with respect to compactness, distance awareness, stability, overlaps, neighbourhood preservation, orthogonal alignment, and user preference. Smaller values for E_c, E_d, E_s, E_o, and closer values to 1 for E_p, E_a and P indicate higher layout quality.

	E_c	E_d	E_s	E_o	E_p	E_a	P
GL	0.318	0.380	0.443	**0**	0.245	$\mathbf{1 - 5.2 \times 10^{-5}}$	0
ST [9]	0.224	0.375	0.392	**0**	0.276	$1 - 6.3 \times 10^{-4}$	0.286
RP [24]	0.192	0.394	0.402	**0**	0.279	$1 - 4.4 \times 10^{-3}$	0
PC [49]	**0**	0.311	0.415	0.310	0.296	$1 - 4.1 \times 10^{-3}$	0
Ours	0.077	**0.198**	**0.106**	**0**	**0.614**	$1 - 2.7 \times 10^{-3}$	**0.714**

They were asked to choose their preferred layout. A subjective criterion P was then employed as the seventh criterion. For each algorithm, this criterion measures the percentage of participants who preferred that algorithm. Smaller values for E_c, E_d, E_s, E_o, and closer values to 1 for E_p, E_a and P indicate higher layout quality. We did not report the results of the non-rectangular boundary (E_r) here because all the methods generated layout results of $E_r = 0$.

Results. Table 2 compares our algorithm with the baseline methods in terms of layout quality. Overall, our layout proved effective as it was the most preferred one among all the algorithms. We also made the following observations from the results:

Compactness and overlapping ratio. Picture collage generates the most space-efficient layout ($E_c = 0$) at the cost of introducing overlapping ($E_o = 0.295$). This is not allowed in our application since overlapping prevents users from clearly seeing the compositions. All the other methods are able to generate layouts without overlaps ($E_o = 0$). Among these methods, our algorithm performs the best in terms of compactness, with 60% less white space on average compared with the other methods. Our algorithm performs better than optimal rectangle packing in terms of compactness because photo resizing is allowed.

Stability, distance awareness and neighbourhood preservation. Our algorithm performed better than the baselines in terms of stability, distance awareness, and neighbourhood preservation. This shows that our algorithm can effectively minimize E_s and E_d. As the layout is aware of composition distance, it naturally preserves the neighbourhood relationships.

Orthogonal Alignment. Both grid layout and squarified treemap algorithms outperformed our algorithm in orthogonal alignment. However, as our algorithm is preferred by most of the participants, we speculate that orthogonal alignment is not important in our application.

6.2 User Study

6.2.1 Study setup

The user study was conducted with 16 photographers. Among the 16 participants, nine (P1-P9) were members of a college photography association, three (P10-P12) were graduate students majoring in photography, and four (P13-P16) were full-time employees of a large IT company. All participants had experience with photo cropping. On average, the participants had 4.13 years of photography experience.

Two major tasks were conducted in the user study:

- T1: Perform single cropping on dataset A.
- T2: Perform batch cropping on dataset B.

At the beginning of each user session, we gave a brief tutorial of the tool. Each task begins with a practice session to familiarize the participant with the task and environment. Then the participants were set free for exploration, working with the photos s/he was interested in. At the same time, the tool logged all the operations. The average time span for the study was 49.5 minutes for each participant.

At the end of the study, each participant was asked to fill in a questionnaire on the usability and the effectiveness of the tool. S/he was also asked to manually crop 5 photos with the tool s/he most frequently used previously, report the time taken to crop the photos, and explain the employed cropping principle.

6.2.2 Results and analysis

We have analyzed the results in terms of cropping time, cropping quality, and usability.

Cropping time. In the single cropping task, the participants spent an average of 14.9s (std=11.4s) to crop each photo with our tool, while they spent an average of 85.2s (std=26.0s) on manually cropping each photo with their frequently used tools.

In the batch cropping task, the average time taken by the participants to browse a group of the top-24 recommendations was 66.5s (std=11.5s). The participants accepted 79% (std=19%) of the top-5 recommendations, which drops to 59% (std=7%) for the top-24 recommendations. In batch cropping, the average time for generating each cropping result was 4.70s, which was much faster than single cropping.

The aforementioned statistics showed that our tool was efficient with both cropping tasks.

Cropping quality. We selected three representative algorithms as the baseline methods, including an interactive and rule-based method [5] (RBM), a fully automatic and learning-based method [55] (LBM), and a thumbnailing algorithm [42] (TM). We compared the cropping results generated by these algorithms with a random result from all three cropping recommendations (PM-R), the cropping recommendations selected by users (PM-S), the final results (PM-F) generated in the user study with our proposed method, and the original photos (ORI).

We invited 27 volunteers to make paired comparisons among the results. For each sample, the user was invited to select which result is preferred or "I find it hard to decide" between the cropping results generated with two different algorithms from the same photo. The preference matrix is presented in Table 3. When generating the cropping recommendations (PM-R), our system required less time than the fully automatic learning-based method (LBM), at the cost of low stability. However, with simple interactions such as selecting a proper recommendation or reference (PM-S), the result was more preferable, and the effect can be further improved with fine-tuning (PM-F). Compared to the other interactive method (RBM), our method is more time-consuming, but generates better results with fewer failures.

Usability. System Usability Scale (SUS) [3] is used to evaluate the usability of PhotoRecomposer. It is a simple, ten-item scale (**I1-I10**) with a global view of subjective assessment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

TABLE 3: Comparison of the cropping results by our proposed method (PM-R, PM-S, PM-F) and baselines (ORI, RBM, LBM, TM) in a preference matrix. The value in a cell in row A and column B indicates the frequency with which a volunteer prefers the result of algorithm A to B, and each cell is calculated with at least 40 samples. Time consumed for the interactive methods includes both time consumed in computation and interactions.

	ORI	RBM	LBM	TM	PM-R	PM-S	PM-F	Time
ORI	-	18.9%	31.2%	54.4%	15.6%	2.3%	1.3%	-
RBM	62.2%	-	43.4%	76.5%	41.8%	14.9%	13.7%	6.5s
LBM	42.5%	29.3%	-	49.3%	36.9%	13.0%	13.3%	43.5s
TM	29.1%	10.6%	24.7%	-	32.5%	10.7%	**13.8%**	41.2s
PM-R	46.9%	12.1%	28.6%	44.2%	-	3.4%	6.2%	**2.7s**
PM-S	68.6%	26.9%	**53.6%**	**84.5%**	**46.9%**	-	5.7%	7.5s
PM-F	**72.5%**	**47.9%**	53.3%	72.5%	**46.9%**	**33.8%**	-	14.9s

of system usability. We used a 5-point Likert scale with 5 = very good usability. Among the items, the participants found the tool was quickly learned (**I7**, mean=4.44, std=0.63), easy to use (**I4**, mean=4.44, std=0.63), consistent (**I6**, mean=4.25, std=0.77). The lowest score was for the item "I think that I would like to use this tool frequently." (**I1**, mean=3.88, std=0.81). The major reason was the participants wanted more functionality (e.g. color balance, resynthesis) in the tool to replace the tools that they usually used.

The total SUS score obtained in the user study was 78.5 ± 14.5, which indicates a good usability for our tool [3].

6.3 Case Study

During the user study, several interesting cases were observed. The cases helped demonstrate the usefulness of single cropping and batch cropping. In addition, we also worked with one photography graduate student (P11) to evaluate example library refinement.

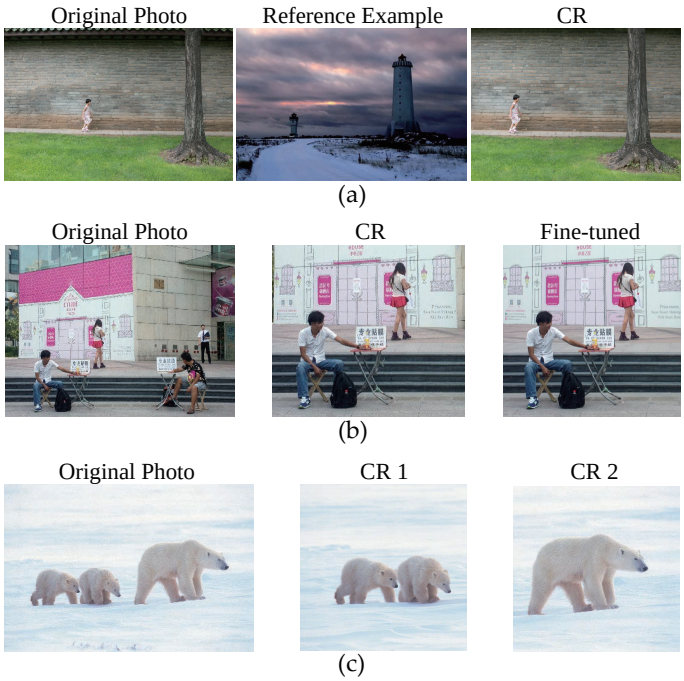


Fig. 8: Cropping recommendations that have (a) better visual balance, (b) unexpected compositions, and (c) a desired aspect ratio. Here CR means cropping recommendation.

6.3.1 Single Cropping

Cropping photos by examining cropping recommendations (R3). PhotoRecomposer helps users better crop a photo by recommending examples that have 1) better visual balance, 2) unexpected but aesthetically pleasing compositions, and 3) a desired aspect ratio.

Improving visual balance. When P15 observed the original photo shown on the left of Fig. 8(a), he quickly realized that this photo is not visually balanced. There is a lot of empty space on the left. As a result, the visual weight towards the right part is too heavy. Normally, he would manually recompose by iteratively cropping the photo and examining whether the cropped version is visually balanced. By using our tool, he quickly checked three cropping recommendations and selected the one that was most visually balanced. The automatic cropping result is shown on the right of Fig. 8(a).

Providing unexpected cropping recommendations. The original photo shown in Fig. 8(b) was taken by P10. He was interested in vendors with small businesses outside the mall. However, the original photo was noisy and none of the vendors appeared to be in focus. He tried to crop the photo so that only two vendors were included, but the result still contained noise and had an undesired aspect ratio. After examining the cropping recommendations, he soon found an unexpected result (Fig. 8(b)) that was satisfying. Although only one vendor was shown, he liked this recommendation because: 1) it showed an interesting comparison between a vendor with a small business outside a mall and a girl who went to the mall for shopping and; 2) the textured area with stairs stood out after the crop. P10 commented, "The textured area is sufficient for attracting attention [13]. The cropping result helps the viewer focus on the vendor sitting over the stairs because of the texture of the stairs." He then manually fine-tuned the result to create a plain and unobtrusive background for the photo (Fig. 8(c)).

Cropping with a desired aspect ratio. P12 liked the original photo shown in Fig. 8(c) very much and wished to upload it as her profile photo on WeChat. Since WeChat requires a profile photo with an aspect ratio of 1:1, she selected "1:1" from the pull-down menu shown in Fig. 1G and examined the corresponding recommendations. Fig. 8(c) shows two recommendations ("CR 1" and "CR 2") provided by our tool, both with desired aspect ratio. After comparing the recommendations, she finally selected "CR 2." She commented, "This cropping makes the subject appear to move through the frame by leaving space in front of the moving object. This follows one basic rule of framing the scene [53]."

Tuning the cropping results by exploring the example library (R1, R4). Although our prototype usually provides good cropping recommendations, sometimes users may still find the recommendations unsatisfying. In such cases, we allow users to improve the cropping result by exploring the example library and selecting better reference examples.

For example, P4 was unsatisfied with the cropping recommendations shown in Fig. 9(a). Although these recommendations removed some unnecessary empty space, they still appeared to be noisy. P4 had no idea how to refine these cropping results and decided to examine the example library for more inspiration. Since the second recommendation was a little better than the others, she selected the second

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

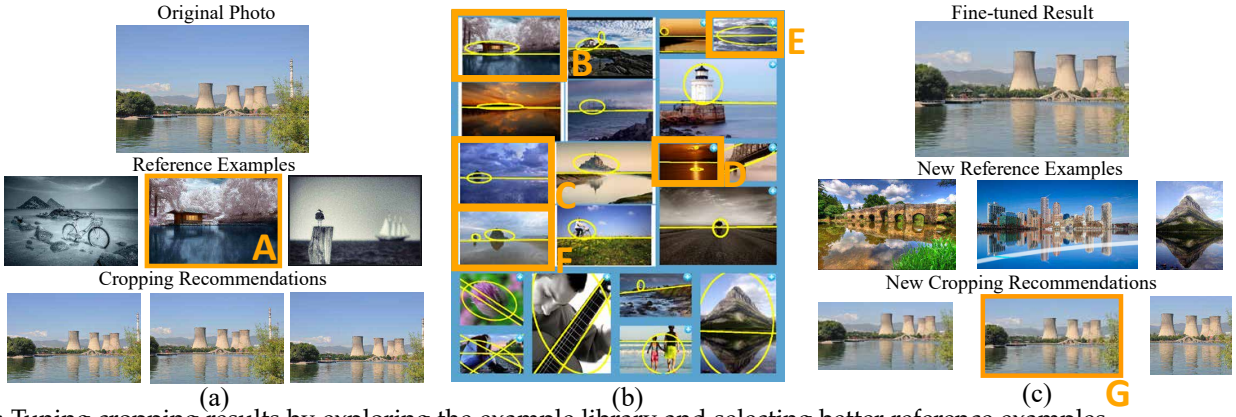


Fig. 9: Tuning cropping results by exploring the example library and selecting better reference examples.

reference example (**A**) as the focus example photo. Our tool generated the layout shown in Fig. 9(b). When P4 examined the example view, some beautiful examples of reflection photography (**C** to **F**) quickly caught her attention. She then realized that she could follow these examples to emphasize captivating reflections in the water by using symmetric composition. Intrigued by this idea, she drilled into cluster **E** to explore more examples. She finally chose the three reference examples shown in Fig. 9(c). After comparing the corresponding cropping results, she quickly decided to use the third one (**G**), and fine-tuned the horizontal visual balance to get the final cropping result.

6.3.2 *Batch Cropping*

This case study demonstrates how PhotoRecomposer helps users efficiently crop a set of photos by using the batch cropping function (**R5**). We used dataset C provided by P5. He took these photos while watching a series of college baseball games. By using the batch cropping function, he was able to generate proper cropping results for 41 photos by carefully cropping only three photos. Here, we only introduce one example of batch cropping.

P5 took many photos of home plate, which is the starting point for much of the action on a baseball field. To generate good cropping results for these photos, he cropped Photo 0 (Fig. 10(b)), which was a typical example of the photos capturing home plate. Photo 0 shows a batter and a catcher on the home plate preparing to hit/catch a ball coming from the right. The cropping box on Photo 0 shows how P5 recomposed this photo. He left more space on the right side, as the empty space helps the viewer imagine the incoming pitch flying through the frame [53]. To make the photo more compact and visually balanced, the space at the bottom of the photo was cut.

After P5 cropped Photo 0, he performed batch cropping and our tool provided cropping recommendations of photos with similar compositions. Photos 1 to 24 in Fig. 10(a) show the top 24 cropping recommendations. These photos were ranked based on their composition similarity with Photo 0. Here the orange boundaries of the recommendations represent the cropping results that were accepted by P5. Overall, the recommendations were accurate, with 16 out of 24 (67%) recommendations accepted. PhotoRecomposer successfully found many photos with similar compositions (e.g., Photos 1-4) and recomposed them by cutting unnecessary space at the bottom or top while leaving empty space on the right. Our tool also worked for photos that had less similar compositions. For example, the subjects in Photos 15, 16, and 22 were much smaller compared with that of Photo 0. Even for these photos, our tool generated proper cropping results by cutting unnecessary empty spaces and giving the subjects a heavier visual weight.

P5 said he usually takes many photos of similar scenes. The batch cropping function enabled him to quickly crop these photos with little effort.

7 Discussion

Although the evaluation demonstrates the usefulness and effectiveness of our prototype, there are several limitations.

First, the performance of our tool depends on the composition structures of photos. It performs well for photos that contain a few salient regions. However, our tool may fail to handle photos with a large number of salient region if the composition extraction method fail to extract the salient regions or the example library does not contain enough similar examples. Typically, the number of relationships, such as the position relationships between salient regions, is exponentially increased with the number of salient regions in a photo. As a result, it usually requires more examples to support the recomposition of such photos. Currently, we solve this problem by providing a manual cropping function.

Second, we currently only consider the composition distance when ranking the recommended examples, which may recommend a different type of example photos (e.g., a scenery photo) to a user photo (e.g., a portrait photo) to be recomposed. After examining the compositions of the two matched photos, the participants said they understood why such photo are recommended first. Several participants commented that recommending different types of photos may trigger them to make an innovative recomposition. In addition, two participants suggested ranking the recommended example by incorporating the information of photo types.

Third, the target users of our tool are amateur photographers. We invited three professional photographers to evaluate PhotoRecomposer. Although they believed that this tool can help them with some simple recomposition cases, two of them pointed out that photo recomposition is an art in most cases, and they prefer to do it manually. As indicated by our user study, the amateur photographers were satisfied with the recomposition results created by our tool.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2764895, IEEE Transactions on Visualization and Computer Graphics

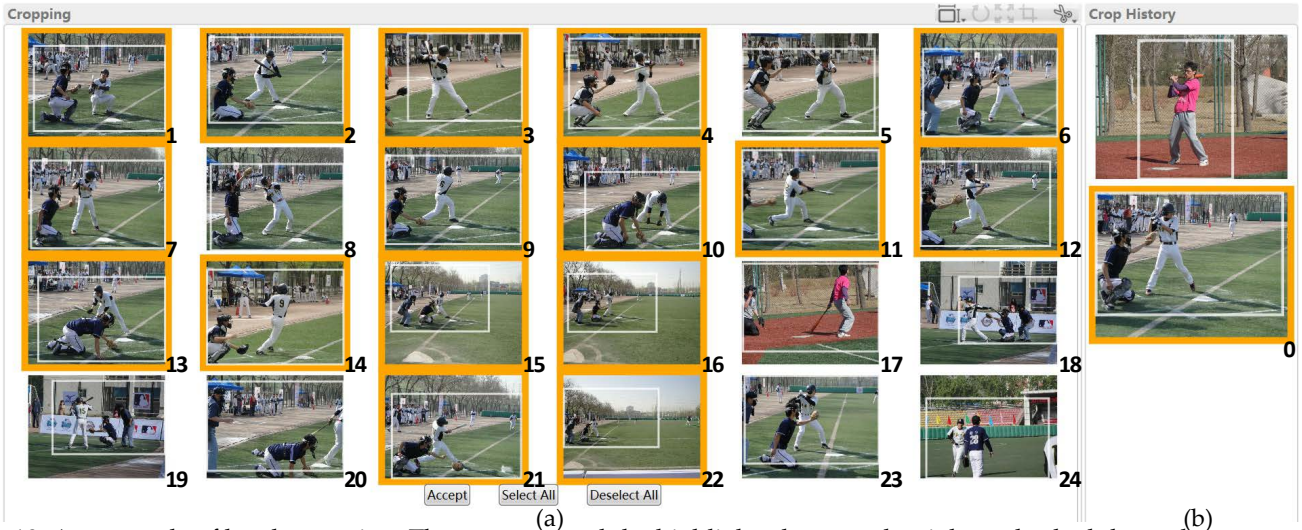


Fig. 10: An example of batch cropping. The user cropped the highlight photo on the right and asked the tool to crop other photos based on this cropping result. The 24 photos with cropping boxes on the left show cropping recommendations provided by our tool. The recommendations accepted by the user were marked in orange.

8 Conclusions and Future Work

We have developed a visual analysis prototype, PhotoRecomposer, to interactively recompose photos. The key aspects of PhotoRecomposer are an earth-mover-distance-based online metric learning algorithm, a multi-level example photo layout method, and a set of interactions for effective recomposition. The experimental results, user study, and case studies demonstrate the efficiency and effectiveness of our prototype in recomposing photos.

We plan to investigate a couple of directions for further research. Our method currently works well for photos that contain a few salient regions. Due to the limitations of the existing composition extraction methods and lack of enough examples, our method may fail to recompose photos with a large number of salient regions. Future directions may include the handling of photos with more salient regions. One possible solution is to employ global image features, e.g., features extracted with a convolutional neural network. We are also interested in incorporating the semantic information of salient regions (e.g., types of salient regions) when ranking the recommended examples, with an aim of recommending the examples with the same type of salient regions first. Finally, we intend to provide more editing functions, such as photo rotation and correcting the composition descriptors, to make the tool more useful.

9 Acknowledgements

This work was supported by the National Key Technology R&D Program (2016YFB1001402), the Natural Science Foundation of China (61672308, 61521002, 61373069), Research Grant of Beijing Higher Institution Engineering Research Center, a Microsoft Research Fund, and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] M. ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28)*, 2015.

[2] M. Balzer and O. Deussen. Voronoi treemaps. In *IEEE Symposium on Information Visualization*, pages 49–56, 2005.

[3] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008.

[4] B. B. Bederson. Photomesa: A zoomable image browser using quantum treemaps and bubblemaps. In *ACM Symposium on User Interface Software and Technology*, pages 71–80, 2001.

[5] S. Bhattacharya, R. Sukthankar, and M. Shah. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 271–280, 2010.

[6] S. Bhattacharya, R. Sukthankar, and M. Shah. A holistic approach to aesthetic enhancement of photographs. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 7S(1):21:1–21:21, 2011.

[7] L. Bottou, J. Weston, and G. H. Bakir. Breaking svm complexity with cross-training. In *Advances in neural information processing systems*, pages 81–88, 2005.

[8] P. Brivio, M. Tarini, and P. Cignoni. Browsing large image datasets through voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1261–1270, 2010.

[9] M. Bruls, K. Huizing, and J. J. Van Wijk. Squarified treemaps. In *Data Visualization*, pages 33–42. 2000.

[10] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[11] J. Chen, G. Bai, S. Liang, and Z. Li. Automatic image cropping: A computational complexity study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 507–515, 2016.

[12] M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.

[13] J. Clements. *Photographic Composition*. Van Nostrand Reinhold Company, 1979.

[14] T. Crnovrsanin, J. Chu, and K.-L. Ma. An incremental layout method for visualizing online dynamic graphs. In *International Symposium on Graph Drawing and Network Visualization*, pages 16–29, 2015.

[15] M. Derthick, M. G. Christel, A. G. Hauptmann, and H. D. Wactlar. Constant density displays using diversity sampling. In *IEEE Symposium on Information Visualization*, pages 137–144, 2003.

[16] D. M. Eler, M. Y. Nakazaki, F. V. Paulovich, D. P. Santos, G. F. Andery, M. C. F. Oliveira, J. Batista Neto, and R. Minghim. Visual analysis of image collections. *The Visual Computer*, 25(10):923–937, 2009.

[17] M. Ester, H. peter Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[18] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[19] C. Fang, Z. Lin, R. Mech, and X. Shen. Automatic image cropping using visual composition, boundary simplicity and content preservation models. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1105–1108. ACM, 2014.

[20] D. Furcy and S. Koenig. Limited discrepancy beam search. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 125–131, 2005.

[21] E. Gomez-Nieto, W. Casaca, D. Motta, I. Hartmann, G. Taubin, and L. G. Nonato. Dealing with multiple requirements in geometric arrangements. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1223–1235, 2016.

[22] Y. Guo, M. Liu, T. Gu, and W. Wang. Improving photo composition elegantly: Considering image similarity during composition optimization. *Computer Graphics Forum*, 31(7):2193–2202, 2012.

[23] X. Han, C. Zhang, W. Lin, M. Xu, B. Sheng, and T. Mei. Tree-based visualization and optimization for image collection. *IEEE Transactions on Cybernetics*, 46(6):1286–1300, 2016.

[24] E. Huang and R. E. Korf. Optimal rectangle packing: An absolute placement approach. *Journal of Artificial Intelligence Research*, 46:47–87, 2013.

[25] J. Huang, H. Chen, B. Wang, and S. Lin. Automatic thumbnail generation based on visual representativeness and foreground recognizability. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 253–261, 2015.

[26] D. F. Huynh, S. M. Drucker, P. Baudisch, and C. Wong. Time quilt: Scaling up zoomable photo browsers for large, unstructured photo collections. In *Extended Abstracts on Human Factors in Computing Systems*, pages 1937–1940, 2005.

[27] J. Kratt, H. Strobelt, and O. Deussen. Improving stability and compactness in street layout visualizations. In *Vision, Modeling, and Visualization*, pages 285–292, 2011.

[28] A. Kuchinsky, C. Pering, M. L. Creech, D. Freeze, B. Serra, and J. Gwizdka. Fotofile: A consumer multimedia organization and retrieval system. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 496–503, 1999.

[29] L. Liu, R. Chen, L. Wolf, and D. Cohen-Or. Optimizing photo composition. *Computer Graphics Forum*, 29(2):469–478, 2010.

[30] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

[31] K. Mizuno, H. Y. Wu, and S. Takahashi. Manipulating bilevel feature space for category-aware image exploration. In *IEEE Pacific Visualization Symposium*, pages 217–224, 2014.

[32] B. Moghaddam, Q. Tian, and T. S. Huang. Spatial visualization for content-based image retrieval. In *International Conference on Multimedia and Expo*, pages 162–165, 2001.

[33] B. Ni, M. Xu, B. Cheng, M. Wang, S. Yan, and Q. Tian. Learning to photograph: A compositional perspective. *IEEE Transactions on Multimedia*, 15(5):1138–1151, 2013.

[34] J. G. Paiva, L. Florian, H. Pedrini, G. Telles, and R. Minghim. Improved similarity trees and their application to visual data classification. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2459–2468, 2011.

[35] H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.

[36] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Evaluating a visualisation of image similarity as a tool for image browsing. In *IEEE Symposium on Information Visualization*, pages 36–43, 1999.

[37] J. B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[38] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. In *ACM transactions on graphics (TOG)*, volume 29, page 160. ACM, 2010.

[39] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen. Gaze-based interaction for semi-automatic photo cropping. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 771–780. ACM, 2006.

[40] P. Simonetto, D. Archambault, and C. Scheideg. A simple approach for boundary improvement of Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):678–687, 2016.

[41] G. Sun, Y. Wu, R. Liang, and S. Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *Journal of Computer Science and Technology*, 28(5):852–867, 2013.

[42] J. Sun and H. Ling. Scale and object aware image thumbnailing. *International journal of computer vision*, 104(2):135–153, 2013.

[43] L. Tan, Y. Song, S. Liu, and L. Xie. Imagehive: Interactive content-aware image summarization. *IEEE Computer Graphics and Applications*, 32(1):46–55, 2012.

[44] X. Tang, W. Luo, and X. Wang. Content-based photo quality assessment. *IEEE Transactions on Multimedia*, 15(8):1930–1943, 2013.

[45] P. van der Corput and J. J. van Wijk. Effects of presentation mode and pace control on performance in image classification. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2301–2309, 2014.

[46] T. van Gog and N. Rummel. Example-based learning: Integrating cognitive and social-cognitive research perspectives. *Educational Psychology Review*, 22(2):155–174, 2010.

[47] R. G. von Gioi, J. Jakubowicz, J. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[48] F. Wang and L. J. Guibas. Supervised earth mover's distance learning and its computer vision applications. In *Proceedings of the European Conference on Computer Vision*, pages 442–455, 2012.

[49] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum. Picture collage. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 347–354, 2006.

[50] X. Wang, S. Liu, J. Liu, J. Chen, J. Zhu, and B. Guo. A full picture of relevant topics. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2508–2521, 2016.

[51] J. Yan, S. Lin, S. B. Kang, and X. Tang. Learning the change for automatic image cropping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 971–978, June 2013.

[52] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *IEEE Symposium On Visual Analytics Science And Technology*, pages 191–198, 2006.

[53] R. D. Zakia and D. A. Page. *Photographic Composition: A Visual Guide*. Focal Press, 2011.

[54] L. Zhang, Y. Gao, R. Ji, Y. Xia, Q. Dai, and X. Li. Actively learning human gaze shifting paths for semantics-aware photo cropping. *IEEE Transactions on Image Processing*, 23(5):2235–2245, 2014.

[55] L. Zhang, M. Song, Q. Zhao, X. Liu, J. Bu, and C. Chen. Probabilistic graphlet transfer for photo cropping. *IEEE Transactions on Image Processing*, 22(2):802–815, 2013.

[56] L. Zhang, Y. Xia, K. Mao, H. Ma, and Z. Shan. An effective video summarization framework toward handheld devices. *IEEE Transactions on Industrial Electronics*, 62(2):1309–1316, 2015.

[57] Canadian online photography community and marketplace 500px. https://marketplace.500px.com/, Jan 2017.

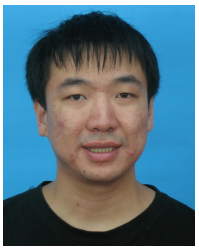
[58] Digital photography challenge website dpchallenge. http://www.dpchallenge.com/, Jan 2017.

[59] Apple iphoto. http://www.apple.com/macos/photos/, Jan 2017.

[60] Adobe photoshop lightroom cc. http://www.adobe.com/products/photoshop-lightroom.html, Jan 2017.

[61] Adobe photoshop cc. http://www.adobe.com/products/photoshop.html, Jan 2017.

[62] Picasa. http://picasa.google.com/, Jan 2017.



Yuan Liang is a PhD candidate in the Department of Science and Technology at Tsinghua University, China. His research interests include interactive multimedia analysis and computer graphics. He received a BS degree in the Department of Science and Technology from Tsinghua University.

Xiting Wang is now an associate researcher at Microsoft Research Asia. Her research interests include text mining and visual text analytics. She received a BS degree in Electronics Engineering from Tsinghua University.

Song-Hai Zhang received the Ph.D. degree from Tsinghua University, Beijing, China, in 2007. He is currently an associate professor of computer science at Tsinghua University. His research interests include image and video processing as well as geometric computing.

Shi-min Hu received the PhD degree from Zhejiang University in 1996. He is currently a professor in the department of Computer Science and Technology, Tsinghua University. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is associate Editor-in-Chief of IEEE Transactions on Visualization and Computer Graphics and The Visual Computer, associate Editor of Computer & Graphics and Computer Aided Design.

Shixia Liu is an associate professor at Tsinghua University. Her research interests include visual text analytics, visual social analytics, interactive machine learning, and text mining. She worked as a research staff member at IBM China Research Lab and a lead researcher at Microsoft Research Asia. She received a B.S. and M.S. from Harbin Institute of Technology, a Ph.D. from Tsinghua University. She is an associate editor of IEEE TVCG and Information Visualization.