Picture Collage

Tie Liu, Jingdong Wang, Jian Sun, Nanning Zheng, *Fellow, IEEE*, Xiaoou Tang, *Fellow, IEEE*, and Heung-Yeung Shum, *Fellow, IEEE*

***Abstract*—In this paper, we address a novel problem of automatically creating a picture collage from a group of images. Picture collage is a kind of visual image summary—to arrange all input images on a given canvas, allowing overlay, to maximize visible visual information. We formulate the picture collage creation problem in a conditional random field model, which integrates image salience, canvas constraint, natural preference, and user interaction. Each image is represented by a group of weighted rectangles, which indicate the salient regions. Then picture collage is resolved by minimizing the energy, guided by the constraints. A two-step optimization method is proposed. First, a quick initialization algorithm based on the proposed 1-D collage method is presented. Second, a very efficient Markov chain Monte Carlo method is designed for the refined optimization. We also integrate user interaction in the formulation and optimization to obtain an interactive collage reflecting personalized preference. Visual and quantitative experimental evaluations indicate the efficiency of the proposed collage creation technique.**

***Index Terms*—1-D collage, interactive collage, Markov chain Monto Carlo (MCMC) optimization, picture collage.**

I. Introduction

With the rapid growth of digital image content, users can feel drowned by the amount of images they come across. For examples, users have to browse hundreds of their vacation photos on a desktop machine, or an image search engine will return thousands of images for a query. To more efficiently view a set of images, image summarization is important to address this problem. Most previous image summarization work mainly focuses on content-based techniques, such as image clustering [1] and categorization [2], to provide a high-level description of a set of images. In this paper, we propose a visual image summarization approach—picture collage. Fig. 1 shows an example of picture collage.

Fig. 1(a) shows a group of images. A common summarization method is to select a smaller number of representative images and create an image mosaic. However, the disadvantage of this approach is that the image mosaic will contain lots of uninformative regions and only few images can be selected. An ideal image summary should contain as many informative regions as possible on a given space. Fig. 1(b) shows a collage produced by a commercial image browsing software. Images are randomly placed on a canvas allowing overlay. Although all images are displayed, more than half of the images are occluded. Additionally, each image is down-sampled, and cropped without considering image content. Fig. 1(c) is a picture collage generated by the proposed approach in this paper. Compared with the previous results, picture collage shows the most informative regions of all images on a single canvas without down-sampling and cropping. In other words, picture collage creates a visual image summarization of a group of images while maximizing visible visual information.

A. Related Work

To create an ideal image summarization, the first step is to determine which regions of each image are informative or salient. Several approaches[3]–[5] on visual attention can output a salience map to indicate the importance of each pixel, while a rectangle enclosing the most salient region is also outputted in [3] and [5]. The approach in [5] defines the local, regional, and global salient features to extract the most salient object and is shown to be superior over [3] and [4]. Thus, in this paper, we adopt it to extract the salient region. In this way, an image is represented by a rectangle enclosing the most salient region.

A simple technique for image arrangement is page layout [6], [7]. It mainly aims to maximize page coverage without allowing image overlap. The "stained glass" photo collage alternatively packs the images together with irregular shapes [8]. However, the two techniques share a common drawback of image mosaics—paying homogenous attention over the entire image.

An interactive approach to combine multiple images is digital photomontage [9]. The user manually specifies salient regions on each image and the system creates a single composite image. This technique works well only when all input images are roughly aligned. But in our application, i.e., picture collage, input images may be completely different.

One generative approach for selecting salient regions and generating a summary image is epitomic analysis [10]. The epitome of an input image is a condensed version of the image

Manuscript received October 20, 2008; revised July 07, 2009. First published August 21, 2009; current version published October 16, 2009. The work of T. Liu and N. Zheng was supported by a grant from the National Science Foundation of China (No.60635050). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Nadia Magnenat-Thalmann.

T. Liu is with the Analytics and Optimization Department, IBM China Research Lab, Beijing 100193, China (e-mail: liutiel@gmail.com; liultie@cn.ibm.com).

J. Wang is with The Media Computing Group, Microsoft Research Asia, Beijing 100190, China (e-mail: jingdw@microsoft.com).

J. Sun is with the The Visual Computing Group, Microsoft Research Asia, Beijing 100190, China (e-mail: jiansun@microsoft.com).

N. Zheng is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: nnzheng@mail.xjtu.edu.cn).

X. Tang is with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: xtang@ie.cuhk.edu.hk).

H.-Y. Shum is with the On-line Service Division, RD, Microsoft, Redmond, WA 98052 USA (e-mail: hshum@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2009.2030741

(a) Image collection (b) Picasa collage (c) Picture collage

Fig. 1. Collages. For all 44 images, the Google's Picasa collage is shown in (b), and a nice collage with more information by our approach is created on a limited canvas in (c).

that contains all constitutive textural and shape primitives necessary for reconstructing the image. But the epitome image is originally designed for the purpose of reconstruction, not for viewing. Semantic structures and objects in the input images cannot be preserved in the epitome image.

The most similar work to ours is digital tapestry [11] and autocollage [12]. Digital tapestry formulates the selection of salient regions and their placement together as a Markov random field (MRF) problem. Each image is represented as a set of blocks, and the multiple-class labeling problem with non-metric constraints is optimized by "truncating" the non-regular energy. However, artifacts are also introduced along the boundaries of neighboring salient regions coming from two different images in digital tapestry, although some artifact removal methods can be used [11]. Autocollage [12] defines different energies to encourage the selection of a representative set of images, select particular object classes, and encourage a spatially efficient and seamless layout. The optimization is divided into a sequence of steps: from static ranking of images, through region of interest detection, optimal packing by the branch-and-bound algorithm, and lastly graph-cut alpha expansion. The core packing algorithm is limited; for example, user interaction cannot be integrated. The packing algorithm cannot deal with images with multiple salient regions which are assigned different weights. Further, the blending still may bring artifacts on the boundaries of different images.

In contrast, picture collage is different from digital tapestry and autocollage in four aspects: 1) Picture collage introduces an overlay style to avoid artifacts caused by the tapestry. This collage style is more common in real life and can often be found in an album designed by artists; 2) The oriented placement and the layer ordering of the image are two unique features in the picture collage. They substantially improve the visual impression of the results. It is not trivial to apply digital tapestry on our picture collage generation; 3) Picture collage is formulated as an energy minimization problem by the conditional random field (CRF) model. All constraints, such as salience, canvas, natural preference, and user interaction, are integrated together, and a two-step optimization is proposed to achieve satisfactory results efficiently; 4) User interaction is integrated in this framework to create a personalized collage.

B. Our Approach

We argue that a nice picture collage should satisfy the following constraints: 1) Salience maximization, salience ratio balance, and no severe occlusion. A picture collage should show as many visible salient regions (without being overlaid by others) as possible. At the same time, each image in the collage has a similar salience ratio (the percentage of visible salient regions). 2) Fitting the canvas. A picture collage should make the best use of the canvas by blank space minimization. If the canvas has an arbitrary shape or a large size, the collage must be as uniform as possible in the canvas. 3) Natural preference. A picture collage must be as natural as possible, which is formulated by spatial uniform, orientation diversity, and layer uniqueness. The spatial distribution is uniform and the orientations of the images are diverse. This property is used to imitate the collage style created by humans. 4) User's interaction. A picture collage should be capable of showing the user's will, while the user's interactions are formulated as the hard or soft constraints on position, orientation, or layer.

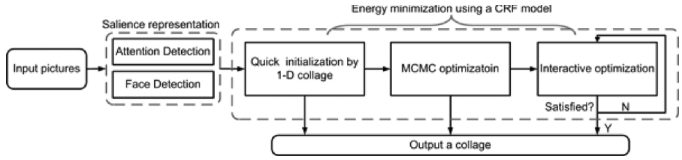
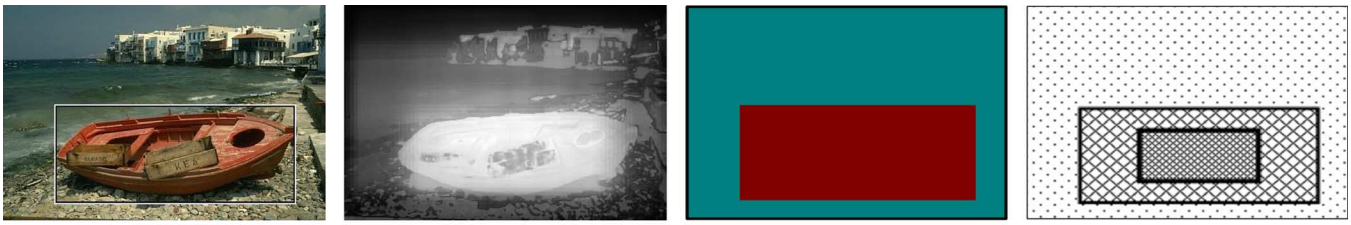


Fig. 2. Picture collage framework. Picture collage is formulated by a CRF model, while quick initialization by 1-D collage, MCMC optimization, and interactive optimization are proposed.

Picture collage is also related to the rectangle packing problem, which is known to be NP-complete [13], [14]. Picture collage is a more challenging problem because of the placement order, and the efficiency is also a key point for such a challenging optimization. Therefore, a two-step optimization method is proposed. Firstly, a quick initialization algorithm, based on 1-D collage, is proposed, while a nice collage can be created from hundreds of images in one second. Secondly, an efficient Markov chain Monto Carlo (MCMC) sampling algorithm is designed for the refined optimization, while very low energy is possible. We also integrate user's interactions into semi-automatic optimization to achieve a satisfactory collage. The whole framework is shown in Fig. 2.

The remainder of this paper is organized as follows. Section II describes the framework of picture collage. The optimization algorithms including quick initialization, MCMC optimization, and interactive optimization are presented in Sections III–V. Section VI presents the experiment and the conclusion is given in Section VII.



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

Fig. 3. Salience representation. (a) Original image, where the rectangle is automatically detected by [5]. (b) Corresponding salient map. (c) One rectangle representation with a linear weight, where the center point of the inside rectangle has a weight 1. (d) Multiple rectangles representation, where pixels in each rectangle have the same weight.

II. Framework

A. Notations

Given N input images $I_{1\cdots N}$ and the salience maps $A_{1\cdots N}$, picture collage arranges all these images on a canvas C. Each image has a set of state variables $X_i = \{s_i, o_i, l_i\}$, where s_i is the 2-D spatial coordinate of image I_i in the canvas, and o_i is the orientation angle. Each image has a unique layer index $l_i \in \{1, 2, \ldots, N\}$ such that we can determine the placement order of all the images in the picture collage. Each image I_i is represented by A_i completely in picture collage. In addition, the constraint from user's interaction is represented as $U = \{U_i\}$. Therefore, our aim is to resolve $X_{1\cdots N}$, given the inputs $A_{1\cdots N}$, C, and U.

B. Salience Representation

The salience map A_i indicates the importance of each pixel from image I_i, and there are several methods [3]–[5], [15] related with salient region extraction. We follow [5], which is shown to be capable to learn the model to detect saliency maps, to compute the salient map by computing the local, regional, and global salient features, which is shown in Fig. 3(b). To be efficient, a rectangle is also resolved to indicate the salience[5]. Usually, the greater weight is assigned to the center of the rectangle, and two kinds of weighting methods are introduced here. First, a linear weight is given to the pixels in the rectangle as in Fig. 3(c), where the center point of the inside rectangle has a maximal weight 1, which will be used at the 1-D collage. Secondly, a set of weighted rectangles is used as in Fig. 3(d), where pixels in each rectangle have the same weight; this type of representation will be used in MCMC optimization. Because the salience map A_i can be represented as multiple weighted rectangles, more factors can be integrated. For example, object recognition can help to discriminate the importance of the regions, and face detection [16], [17] has been studied extensively. The rectangles from the face detector [16] are represented as the same as the salience rectangles. A larger weight is assigned to the face rectangles because all faces are expected to be visible, especially in family photos. Users can also assign an important region in an image by hand, where the region is also represented as a weighted rectangle. We represent these weighted rectangles and the canvas as polygons. Then, the computation, such as overlaps and occlusions, during the arrangement can be performed efficiently by very simple polygon boolean operations.[1]

C. Problem Formulation

With the conditional random field framework [18], picture collage can be formulated as a conditional distribution:

$$P(X_{1\cdots N}|D) = \frac{1}{Z}\exp\left(-\sum_{i=1}^{N} E_i(X_i|D, \overline{X}_i)\right) \quad (1)$$

where $D = \{A_{1\cdots N}, C, U\}$ are the given inputs, $\overline{X}_i = X_{1\cdots i-1, i+1\cdots N}$ are the constraints from the arrangements of other images, and Z is the partition function. Compared with Markov random field, one of the advantages of conditional random field is that the constraints from all observations can be integrated in the conditional probability. Maximizing the probability is equivalent to minimizing the sum of energies, and the optimal states for picture collage can be resolved:

$$X^*_{1\cdots N} = \arg\min_{X_{1\cdots N}} \sum_{i=1}^{N} E_i(X_i|D, \overline{X}_i). \quad (2)$$

A reasonable supposition is that these constraints are conditionally independent, and the energy can be decomposed as

$$\sum_{i=1}^{N} E_i(X_i|D, \overline{X}_i) = \underbrace{E_S(X_{1\cdots N}, A_{1\cdots N})}_{\text{salience constraint}} + \underbrace{E_C(X_{1\cdots N}, C)}_{\text{canvas constraint}} + \underbrace{E_P(X_{1\cdots N})}_{\text{natural preference}} + \underbrace{E_U(X_{1\cdots N}, U)}_{\text{user's interaction}}. \quad (3)$$

Each constraint tries to simulate one aspect in a human's activities for collage, and is described as follows.

D. Salience Constraint

The salience constraint measures the cost from visible salience with given salience map $A_{1\cdots N}$, and includes salience maximization, salience ratio balance, and penalty of severe occlusion:

$$E_S(X_{1\cdots N}, A_{1\cdots N}) = \overline{A}_{occ} + \lambda_V V_A + \lambda_O O_A \quad (4)$$

[1]A fast implementation can be obtained from http://www.cs.man.ac.uk/ toby/alan/software/gpc.html.

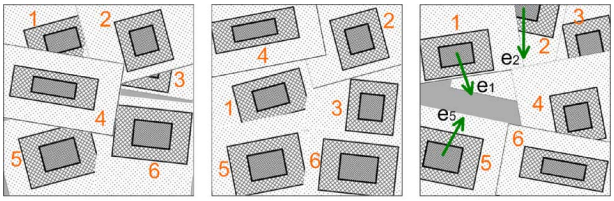


Fig. 4. Salience constraint. In (a), most of image 3 is invisible. With the salience ratio balance criterion a collage with the same images is better balanced as shown in (b). In (c), the directions e_1 and e_5 are computed for image 1 and image 5 from blank space. The direction e_2 is computed for image 2 from moveable space, respectively.

where $\overline{A}_{occ}$ is the normalized sum of occluded salience regions, V_A is the variance of salience ratios, and O_A is the penalty of severe occlusion.

1) Salience Maximization: This property aims to maximize the total amount of visible salience $A_{vis} = \sum_i A_i^{vis}$, where A_i^{vis} is the visible part of the salience region A_i and it can be computed quickly with the help of rectangle representation and polygon operation. Salience maximization is equivalent to minimizing the sum of occluded salience regions $A_{occ} = A_{\max} - A_{vis}$, where $A_{\max} = \sum_i A_i$. We further normalize this measure in the range $[0, 1]$:

$$\overline{A}_{occ} = \frac{A_{occ}}{A_{\max}}. \tag{5}$$

2) Salience Ratio Balance: Due to canvas size limitation, the visible part of a salience region may be very small as image I_3 in Fig. 4(a). To avoid such a result, a visible salience ratio balance can be introduced to obtain a well-balanced collage as shown in Fig. 4(b). The visible salience ratio of one image is calculated as $r_i = A_i^{vis}/A_i$. The variance of all visible salience ratios

$$V_A = \frac{1}{N}\sum_{i=1}^{N}(r_i - \bar{r})^2 \tag{6}$$

is used to evaluate the balance, where $\bar{r} = \sum_{i=1}^{N} r_i/N$. A well-balanced picture collage tends to have a smaller variance V_A.

3) Penalty of Severe Occlusion: Salience ratio balance cannot guarantee that all images are visible, especially when they are influenced by other constraints. To make sure that each image is partly visible, the penalty of severe collusion is defined as

$$O_A = \sum_{i=1}^{N}\delta\left(\frac{A_i^{vis}}{A_i} < t_O\right) \tag{7}$$

where $\delta(x) = 1$ if x is true, and 0 otherwise. $t_O = 0.1$ is the threshold that an image is occluded severely.

To balance the relative importance of the above factors, we set the weight $\lambda_V = 2$, and $\lambda_O = 100$ which gives a big penalty if an image is occluded severely. If λ_O is too small, there may be some images occluded severely, especially when the number of images is large. These two parameters are important to avoid salient region occlusion.

E. Canvas Constraint

The canvas constraints measure the cost from the given canvas C. These are blank space minimization and canvas shape constraint:

$$E_C(X_{1\cdots N}, C) = \lambda_B \overline{B} + \lambda_M C_M \tag{8}$$

where $\overline{B}$ is the normalized sum of uncovered regions on the canvas, and C_M comes from the canvas shape constraint.

1) Blank Space Minimization: The blank space is the space in the canvas that is not covered by any image. The blank space can be calculated as the difference of the canvas bounding polygon R_C and the union of all the images: $B = R_C - \bigcup_{i=1}^{N} R_i$, where R_i is the bounding polygon of image I_i. B should be minimized to make the best use of canvas space. We also compute the normalized term

$$\overline{B} = \frac{Area(B)}{Area(R_C)}. \tag{9}$$

2) Canvas Shape Constraint: The canvas can have arbitrary shapes, where the images are only arranged in the valid region. The binary shape mask $M \in \{0, 1\}$ indicates where the image can be arranged, and it can be fitted as a polygon P_M. Then, the cost is defined as

$$C_M = \sum_{i=1}^{N}\delta(s_i \notin P_M) \tag{10}$$

where s_i is the ith image's position. It penalizes the image not arranged in the canvas.

To balance the relative importance with other factors, we set the weight $\lambda_B = 10$, and $\lambda_M = 1000$ which indicate that the canvas shape is almost a hard constraint. If λ_B is too small, there may be blank areas, especially when the number of images is small.

F. Natural Preference

Naturally speaking, a visually pleasing collage must be spatially uniform, orientationally diverse, and layer exclusive. We factorize the prior $E_P(X_{1\cdots N})$ in (3) as

$$E_P(X_{1\cdots N}) = \lambda_{P_S} P_S + \lambda_{P_O} P_O + \lambda_{P_L} P_L \tag{11}$$

where P_S, P_O, P_L represent spatial uniformity, orientational diversity, and layer exclusion, respectively, and the weights are all set to 10.

1) Spatial Uniformity: All images are expected to be arranged uniformly on the canvas. That means each image has almost similar distance to their neighbors. Suppose image pairs are neighbors: (s_i, s_j), the variance of distance between neighboring images should be minimized:

$$P_S = \frac{1}{bN}\sum_{i=1}^{N}\sum_{j\in N_i}(M_s - d(s_i, s_j))^2 \tag{12}$$

where $M_s = (1/bN)\sum_i \sum_{j\in N_i} d(s_i, s_j)$ is the mean distance, $b = 4$ is the number of neighbors and N is the number of images, and $d(s_i, s_j)$ is the normalized spatial distance between s_i

(a) (b) (c)

Fig. 5. 1-D collage. (a) Original images with detected salient rectangles, (b) 1-D collage without rotation, and (c) 1-D collage with rotation.

and s_j. Each image is supposed to have four neighbors, which can be found along for different orientations.

2) Orientational Diversity: We observed that the collage would look very stiff if all the images are arranged in the same orientation. To make the collage visually attractive, we bring an orientational diversity term to expect that the images are arranged with diverse orientations. To globally measure the orientational diversity of all images, we compute the average of the absolute orientation difference between any two images I_i and I_j: $o_g = \sqrt{\sum_{i=1,j=1}^{N} |o_i - o_j|^2/(N(N-1))}$. To encourage orientational diversity, we model P_O as

$$P_O = -\log\left(N(o_g; m_g, \sigma_g^2)\prod_{i=1}^{N} N(o_i; 0, \sigma_o^2)\right) \tag{13}$$

where $\{m_g, \sigma_g\}$ controls the global diversity of orientations. The second term $N(o_i; 0, \sigma_o)$ encourages each image to be placed upright, where the variance σ_o controls the individual diversity.

3) Layer Exclusion: To compute the visible saliency map, it is necessary to obtain the overlapping relations between all pairs of images, which can be determined by the layer indices of the images. To get the overlapping relations conveniently, we propose a unique layer index constraint, that is, each image has a different index. The unique layer index constraint is a strict constraint since it can get the same overlapping relations to allow that two images have the same layer index. However, it would help formulate and solve the problem without changing the final result. In order to assign a unique layer index to each image, we model P_L as

$$P_L = -\log\left(\frac{\prod_{i=1}^{N}\prod_{j\neq i}\delta(l_i, l_j)}{n!}\right) \tag{14}$$

where $\delta(l_i, l_j)$ is an indicator function defined above.

G. User's Interaction

People can operate the position, orientation, and layer for each image and each operation generates a preference such as s_{iU}, o_{iU}, l_{iU}. Suppose the set of images N_U are operated, and $E_U(X_{1\cdots N}, U)$ in (3) can be factorized as

$$E_U(\cdot) = \sum_{i\in N_U}(\lambda_s E_U(s_i, U) + \lambda_o E_U(o_i, U) + \lambda_l E_U(l_i, U)) \tag{15}$$

where $E_U(s_i, U), E_U(o_i, U), E_U(l_i, U)$ are the costs from the operation on position, orientation, and layer. Suppose that the position preference is the Gaussian distribution with the assigned position s_{iU}, then $E_U(s_i, U) = -\log(N(s_i; s_{iU}, \sigma_s^2))$, where σ_s is the allowable variance for position's preference; $\sigma_s = 0$ denotes the hard constraint. The orientation distribution is also a Gaussian distribution with the given o_{iU}, but with a smaller variance: $E_U(o_i, U) = -\log(N(o_i; o_{iU}, \sigma_o^2))$. A unique label is assigned to each layer, and layer preference is a hard constraint with the zero variance: $E_U(l_i, U) = -\log\delta(l_i, l_{iU})$. To prefer the user's interaction, $\lambda_s, \lambda_o, \lambda_l$ are all set to 100.

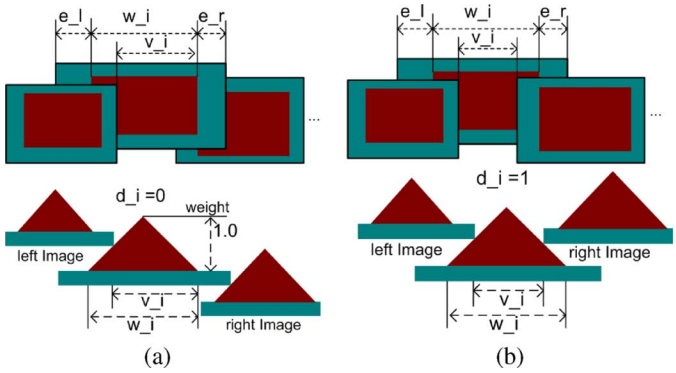


Fig. 6. 1-D collage along the horizontal axis. (a) Occluded by the left image. (b) Occluded by the left and right images. The top row is the top view, and the bottom row is the lateral view. The carmine region is the salient region, and a linear weight is given to the salient rectangle while the center has the maximal weight 1.0. v_i is the visible salient rectangle, w_i is the width of salient rectangle, e_l, e_r are the left and right boundaries of image. $d_i \in \{0, 1\}$ means whether A_i is occluded by the right image A_{i+1}.

H. Optimization

To minimize such an energy efficiently, a two-step optimization method is proposed. Firstly, a very efficient initialization algorithm based on 1-D collage is presented to create an initial collage. Secondly, an efficient MCMC optimization algorithm is presented to refine the collage. A semi-automatic collage is also designed while the user's interactions are integrated. The whole optimization strategy is shown in Fig. 2, and we will deploy them, respectively, in the following.

III. Quick Initialization by 1-D Collage

The quick initialization algorithm is based on a simplified problem: 1-D collage, which means to arrange the images in one dimension and can be optimized determinatively with high efficiency. This algorithm can generate a nice collage from hundreds of images in seconds.

A. 1-D Collage

As shown in Fig. 5(a), the salient rectangles $A_{1\cdots N}$ are automatically detected from the images, and they are arranged along the horizontal line as in Fig. 5(b) and (c), given the order of images and the canvas width W. In Fig. 6(a), for each image A_i along the row, w_i indicates the width of the salient rectangle, and e_{li}, e_{ri} are the left and right widths surrounding the salient

region. The whole width of the image is $w_i + e_{li} + e_{ri}$. The unknown variables are the positions of each of the salient rectangles s_i, the layers l_i, and the orientation o_i, where s_i is the horizontal coordinate of the center of the salient rectangle. We define two extra variables:

- $v_i \in [0, w_i]$ is the visible salient rectangle width, while w_i is the width of the salient rectangle;
- $d_i \in \{0, 1\}$ indicates whether or not A_i is occluded by the right image A_{i+1}.

We propose a sequential optimization method to this 1-D collage problem: position and layer optimization. We will prove that the final positions and layers $\{s_i, l_i\}$ can be resolved from $\{v_i, d_i\}$ determinately given the image order, and we will involve the layer uniqueness constraint in the optimization procedure as a hard constraint, which will be omitted from the optimal objective function. The main energy constraint comes from the salience constraint, and the simplified energy optimization is written as

$$\{v_i, d_i\}^* = \arg\min_{\{v_i, d_i\}} \sum_{i=1}^{N} ((1 - r_i) + \lambda_O \delta(r_i < t_O) + \lambda_V (r_i - \bar{r})^2), \quad (16)$$

where $r_i = A_i^{vis}/A_i$ is the ratio of A_i^{vis} and A_i, and A_i^{vis} indicates the visible salient region. It can be observed that only the ratio r_i is involved in this equation; hence, it is sufficient to represent the salient and visible salient regions according to their widths. Moreover, particularly in 1-D collage, the weight in the salient rectangle is approximated in a linear decreasing function with respect to the distance of a pixel from the center of the salient rectangle. Thus, $A_i = w_i/2$ indicates the whole salient region. Furthermore, $\bar{r} = (1/N)\sum_{i=1}^{N} A_i^{vis}/A_i$ is denoted as the mean salience ratio. For $\bar{r}$ is a global variable which is computed from all A_i^{vis}, the optimization can be resolved using the expectation-maximization manner: first, we minimize the objective function to resolve A_i^{vis}, v_i, d_i using the previous computed $\bar{r}$; second, we compute $\bar{r}$ using the previous computed A_i^{vis}, v_i, d_i. At the first iteration, the third item containing $\bar{r}$ in (16) is omitted. Usually, the optimization can be converged in three to five iterations.

As shown in Fig. 6, if A_i^{vis} can be computed from v_i, d_i, d_{i-1}, the optimal v_i, d_i in (16) can be resolved determinately. We can observe that when A_i is occluded by the left and right images, the optimal position of A_i is at the center as shown in Fig. 6(b). If A_i is above the left and right images ($d_i = 0, d_{i-1} = 1$), then $v_i = w_i$. The concluded equation is

$$A_i^{vis}(v_i, d_{i,i-1}) = \begin{cases} \frac{w_i}{2} - \frac{(w_i - v_i)^2}{(2 * w_i)} & d_i = 1, d_{i-1} = 0 \\ \frac{w_i}{2}, & d_i = 0, d_{i-1} = 1 \\ \frac{v_i^2}{w_i}, & d_i = d_{i-1}, v_i \le \frac{w_i}{2} \\ \frac{w_i}{2} - \frac{(w_i - v_i)^2}{w_i}, & d_i = d_{i-1}, v_i > \frac{w_i}{2}. \end{cases} \quad (17)$$

A virtual point $d_0 = 0$ supposes that the first image is occluded from the left, and $d_N = 1$ indicates that the Nth image is occluded from the right.

We can resolve the optimal $\{v_i, d_i\}$ by dynamic programming. Suppose that image A_i is added at time clock i along the line from left to right, and the current canvas width is W_i. d_i is also considered as a state variable to indicate whether the current image will be occluded by the next image. The decision variable is v_i. Then the optimal value function $T_i(W_i, d_i)$ can be defined as

$$T_i(W_i, d_i) = \arg\min_{\{v_i, d_{i-1}\}} (T_{i-1}(W_{i-1}, d_{i-1}) + V_i) \quad (18)$$

where V_i is the objective function for the ith image that is minimized in (16), and W_{i-1} is computed as

$$W_{i-1} = \begin{cases} W_i - (v_i + e_{li}), & d_{i-1} = 1 \\ W_i - (v_i + e_{ri-1}), & d_{i-1} = 0 \end{cases} \quad (19)$$

where e_{li}, e_{ri} are the left and right boundary widths as shown in Fig. 6. The initial condition is $T_0(0, d_0) = 0$, and $d_0 = 0$ as mentioned above. The final optimal resolve can be inferred from $T_N(W, d_N)$, where W is the whole width of the canvas, and $d_N = 1$ means the most right image is always occluded by the canvas frame. When the images cannot be arranged in the canvas without occlusions of salient regions: $W \le \sum_{i=1}^{N} w_i + \sum_{i=1}^{N-1} \min(e_{li}, e_{ri+1})$, the canvas must be filled in and backward from $T_N(W, d_N)$ promises that the salient regions are maximized and there is no blank space on the canvas. Otherwise, an easier method can arrange the images without dynamic programming: the space between two salient regions is set as $(W - \sum_{i=1}^{N} w_i)/(N - 1)$. This case with a large canvas can be also extended to spatial collage, where the prior from spatial uniform and orientation diversity will play the main functions and where other energies will remain unchanged after several iterations when there are no overlaps between images.

Then $\{s_i, l_i\}$ can be resolved through $\{v_i, d_i\}$. To be clear, the start position of each visible salient rectangle P_i is computed as an intermediate variable:

$$P_i = \begin{cases} P_{i-1} + v_{i-1} + e_{li}, & d_{i-1} = 1 \\ P_{i-1} + v_{i-1} + e_{ri-1}, & d_{i-1} = 0 \end{cases} \quad (20)$$

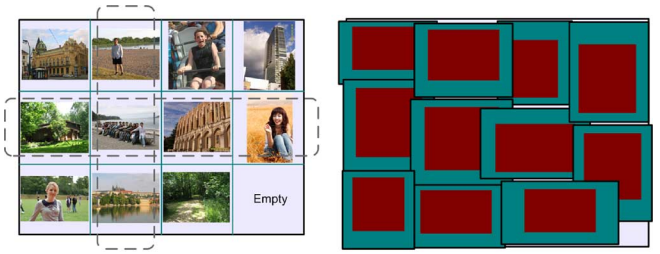
where $P_1 = 0$. Then the center of the salient rectangle s_i (the horizontal coordinate here) is computed:

$$s_i = \begin{cases} P_i + \frac{w_i}{2} & d_{i-1} = 1 \\ P_i + v_i - \frac{w_i}{2} & d_{i-1} = 0, d_i = 0 \\ P_i + \frac{v_i}{2} & d_{i-1} = 0, d_i = 1. \end{cases} \quad (21)$$

The layers $\{l_i\}$ can be resolved using a bi-directional chain: if $d_i = 0$, the next image A_{i+1} should be put below A_i, and vice versa. Then $\{l_i\}$ is assigned as $1 \cdots N$ from the lowest image in the chain. Orientation o_i is resolved subsequently with the orientational diversity constraint, and the result is shown in Fig. 5(c). Now, all the state variables $\{s_i, l_i, o_i\}$ are approximately optimally resolved. We can also resolve the vertical coordinates along each column with the same strategy, and the dynamic programming can be speeded up through down-sampling.

B. Decompose into 1-D Collage

Although 1-D collage can be resolved efficiently using the simplified energy, can the spatial collage be decomposed into the vertical and horizontal 1-D collage? The intuitive thought is



(a) (b)

Fig. 7. Decompose into 1-D collage. (a) Computed grids, the bottom-right grid being empty. Dashed lines mark out the 1-D collage along the row and the column. (b) Assign an image for each grid, to be as uniform as possible.

to divide the canvas into grids with N_r rows and N_c columns, where $N_r \approx \sum_i w_i/W, N_c \approx \sum_i h_i/H$ and $N_r \times N_c \geq N$. Each grid is assigned an image $A_{n_{kl}}$ as in Fig. 7(a), where $n_{kl} \in [1, N]$ is the image index and $k \in [1, N_r], l \in [1, N_c]$ are the kth row and lth column.

Suppose images are arranged in the canvas as uniform as possible, all rows of canvas grids have almost a similar sum of "salience width": $M_r = \sum_i w_i/N_r$. Similarly, all columns have a similar sum of "salience height": $M_c = \sum_i h_i/N_c$. Then the optimal indices can be resolved:

$$\{n_{kl}\}^* = \arg\min_{\{n_{kl}\}} \left(\sum_{k=1}^{N_r} \left(M_r - \sum_{l=1}^{N_c} w_{n_{kl}} \right)^2 + \sum_{l=1}^{N_c} \left(M_c - \sum_{k=1}^{N_r} h_{n_{kl}} \right)^2 \right). \tag{22}$$

The optimization can be achieved by the following two steps:

1) Grouping by Rows: The row indices are assigned to minimize $L_r = \sum_{k=1}^{N_r} (M_r - \sum_{l=1}^{N_c} w_{n_{kl}})^2$, which is a typical integer programming problem. A greedy method is used to select an image for the kth row at each time to minimize $(M_r - \sum_{l=1}^{N_c} w_{n_{kl}})^2$. The cropped dynamic programming algorithm can speed up the selection, and the selection begins from the last row. After all rows are assigned, image pairs are switched between rows to minimize L_r: a pair of images from rows with the minimal and maximal salience widths are switched if L_r decreases. This progress is repeated until L_r does not decrease.

2) Adjustment Between Columns: After the images are grouped by rows, their columns are adjusted to minimize $L_c = \sum_{l=1}^{N_c} (M_c - \sum_{k=1}^{N_r} w_{n_{kl}})^2$: similarly, an image pair is searched to be switched to minimize L_c. This operation is repeated until L_c does not decrease any more.

Now, all the images are arranged in the grids to make them as uniform as possible, as shown in Fig. 7(b). Then 1-D collage can be resolved along each row and each column, respectively. The spatial position s_i is composed with the horizontal and vertical coordinate. The spatial layer l_i can be computed from the resolved layer in 1-D collage. The orientation o_i can be resolved independently using the orientational diversity constraint.

C. Analysis on Quick Initialization

Two examples from quick initialization are shown in Fig. 8(a) and (b). Using the simplified energy, quick initialization can achieve a nice collage for a large number of images as in Fig. 8(a). When the number of images decreases, there may be artifacts such as blank canvas and occlusions as in Fig. 8(b). This is because only one dimensional constraint along the row or the column is considered during 1-D collage. As proved in experiments, a quick initialization algorithm is quite efficient, and it can output a nice collage from 500 images in seconds and improve the efficiency of the whole collage optimization dramatically. For those artifacts from quick initialization, such as shown in Fig. 8(b), the MCMC optimization in the following section considering all energies from (3) is proposed to refine the collage.

(a) (b)

Fig. 8. Results from quick initialization. (a) is with 32 images, and (b) is with 14 images taking less than 0.05 s. Quick initialization can output a nice result from a large number of images as in (a). When the number of images decreases, there may be artifacts such as the occlusion in (b). MCMC optimization can be used to refine the result as in (b).

IV. MCMC Optimization

In this section, we present a Markov chain Monte Carlo algorithm to refine the quick initialized collage. MCMC is a powerful sampling-based optimization method to the complex optimization problems [e.g., (3)] that cannot be solved by some standard numerical optimization techniques (e.g., gradient-based methods). It has been shown that a good initialization (i.e., initial collage parameters including positions, layer indices, and orientations of the images) can speed up the MCMC optimization convergence. The collage parameters obtained in quick initialization by 1-D collage are used as the initial parameters of the MCMC optimization. In the following, we present details on how to design the proposals.

A. Markov Chain Monte Carlo (MCMC)

Given a distribution $\pi(X)$ of variables X, in our case $X = \{s_i, o_i, l_i\}$, MCMC is a strategy for generating samples $\{X^k\}_{k=1}^K$ of $\pi(X)$ by exploring the state space of X using a Markov chain mechanism. This mechanism constructs a chain that spends more time in the regions with higher probability density. The stationary distribution of the chain will be the target distribution $\pi(X)$. In this paper, we pick the sample

$$X^{k^*} = \text{argmax}_{X^k} p(X^k|D)$$

as the MAP solution, and $D = \{A_{1...N}, C, U\}$ are the inputs as mentioned above.

Most MCMC methods are based on the *Metropolis-Hastings* (MH) algorithm. In MH sampling, the proposal function $q(X^*|X^k)$ (also called transition kernel) can be an arbitrary distribution that is used to sample a candidate sample X^* given the current state X. It is the key factor that affects sampling

efficiency. In other words, whether or not an MCMC approach can effectively sample the target distribution $\pi(X)$ completely depends on how well the proposal function $q(X^*|X^k)$ is designed.

B. Proposal Design

There must be many local optimums in our high dimensional, non-convex combinatorial optimization problem. To avoid sticking at local minima, we design a mixture of proposals to deal with this problem: 1) a local proposal q_l that discovers finer details of the target distribution, 2) a global proposal q_g that can explore vast regions of the state space of X, and 3) a pairwise proposal q_p that has the property in between. The mixture of proposal $q(X^*|X^k)$ is defined as

$$q(X^*|X^k) = v_l q_l(X^*|X^k) + v_g q_g(X^*|X^k) + v_p q_p(X^*|X^k) \tag{23}$$

where v_l, v_g, and v_p are three weights which will be dynamically adjusted. Both global and pairwise proposals are critical to make the algorithm jump out from a local minimum.

For clarity, let $\pi(X_i, *) \equiv \pi(X_i, X \setminus X_i)$ when only the state X_i is involved for update. Similarly, $\pi(s_i, *) \equiv \pi(s_i, X \setminus X_i)$ when only the position of image I_i is involved.

1) Local Proposal: Local proposal only changes the state of one image once. The proposal $q_l(X^*|X^k)$ should determine: 1) which image is to be selected for update and 2) how to propose a good state for the selected image in a probabilistic manner.

For the first issue, we compute a weight w_i for each image. This weight is inversely proportional to the visible saliency ratio $r_i = A_i^{vis}/A_i$ of each image:

$$w_i = \frac{(r_i + \epsilon)^{-1}}{\sum_i (r_i + \epsilon)^{-1}} \tag{24}$$

where $\epsilon = 0.2$ is a constant to dilute the influence of this weighting. We select the image I_i with the probability w_i.

For the second issue, the most frequently seen method is random walk sampling, i.e., adding a random disturbance to the current state configuration. However, in random walk sampling, it is often the case that a small step-size in the proposal will result in exceedingly slow movement of the corresponding Markov chain, whereas a large step-size will result in a very low acceptance rate. To avoid such "blind" sampling, we propose the following sampling algorithm to make large step-sizes without lowering the acceptance ratio based on the Multipoint Metropolis method [19].

To update the state $X_i = \{s_i, o_i, l_i\}$, we randomly select one of following proposals: position proposal, orientation proposal, and layer proposal.

Position Proposal: Our position proposal is based on Random-Grid Sampling [20] (RGS).

- Randomly generate a direction e and a gird size r.
- Construct the candidate set as

$$y_m = s_i^k + m \cdot r \cdot e,\ m = 1, \ldots, M.$$

- Draw y from $\{y_m\}_{m=1}^M$ with probability $\pi(y_m)$.
- Construct the reference set $\{y_m^r = y_m - m \cdot r \cdot e\}_{m=1}^M$.
- Let $s_i^{k+1} = y$ with probability

$$\min\left\{1, \frac{\sum_{m=1}^M \pi(y_m)}{\sum_{m=1}^M \pi(y_m^r)}\right\}$$

or reject otherwise.

Conceptually, RGS performs a 1-D probabilistic search on a random direction such that it can make a large step-size jump from the current state. However, the random sampling of the direction in RGS is still blind. Therefore, we should propose the direction e so that it has more space—either blank space driven RGS or "moveable" direction driven RGS.

Blank Space Driven RGS: Given a current state configuration, there may be a number of blank regions. In the case there is at least one adjacent blank region B_i for image I_i (we randomly select one if there is more than one adjacent blank region), we obtain a direction e_i^B from the center of the bounding rectangle R_i to the center of the union region $R_i \bigcup B_i$, e.g., the direction e_3 of image 3 in Fig. 4(c). Then we sample e and r for RGS from two Gaussian distributions $N(e; e_i^B, \sigma_e^2)$ and $N(r; m_r, \sigma_r^2)$, respectively. This proposal is in particular useful in the early phase of the sampling when there are many blank regions. We set $M = 10$ in RGS.

Moveable Direction Driven RGS: In the case there is no adjacent blank region for image I_i, we consider directions $\{e_j\}_{j=1}^n$ from its center to the centers of its n adjacent images $\{I_j\}_{j=1}^n$. First, we denote R_i^a as the saliency bounding rectangle of the saliency region in image I_i. Second, we define a "moveable" distance d_{ij} between image I_i to its neighbor I_j. If the image I_i is on the top of the image I_j, the "moveable" distance d_{ij} is the minimal distance between the bounding rectangle R_i of image I_i and the saliency bounding rectangle R_j^a of image I_j [e.g., from image 4 to image 5 in Fig. 4(c)]; otherwise, d_{ij} is the minimal distance between R_i^a and R_j [e.g., from image 5 to image 4 in Fig. 4(c)]. Lastly, we sample a direction e_i^M from the direction set $\{e_j\}_{j=1}^n$ with the probability that its value is proportional to $\{d_{ij}\}_{j=1}^n$. The final direction e for RGS is again sampled from a Gaussian direction $N(e; e_i^M, 6\sigma_e^2)$. In the case all the distances d_{ij} are 0, a random direction is sampled. This proposal is quite useful in the whole phase of the sampling.

Orientation Proposal: The RGS method can be directly applied on orientation proposal because the orientation o_i is a 1-D variable. Direction sampling is not necessary. We sample a grid size r from a Gaussian distribution $N(r; \sigma_o/5, \sigma_o^2/20)$. M is also set as 10.

Layer Proposal: To sample layer index l^{k+1}, we do not need to consider the layer index l^k because layer change will often cause a large change of the likelihood. Therefore, we generate the layer index using Multiple-Try Metropolised Independence Sampling (MTMIS) [20]. Its basic process is as follows.

- Uniformly draw a trail set of layer index samples $\{y_m\}_{m=1}^M$ for the set $\{1, 2, \ldots, N\}$. Compute $W = \sum_{m=1}^M \pi(y_m, *)$.
- Draw a layer index y from the trail set $\{y_m\}_{m=1}^M$ with probability proportional to $\pi(y_m, *)$.
- Let $l^{k+1} = y$ with probability

$$\min\left\{1, \frac{W}{(W - \pi(y, *) + \pi(l^k, *))}\right\}$$

and let $l^{k+1} = l^k$ otherwise.

(a) (b)

Fig. 9. MCMC optimization. (a) is with 19 images, and (b) is with 22 images.

We set the number $M = 2N$ so that we have a good chance to search a better layer index in a probabilistic manner. Another big advantage of using multiple-try sampling is that we can precompute $\{\pi(l_i = 1, *), \ldots, \pi(l_i = N, *)\}$ incrementally such that the computation cost of multiple-try sampling is just twice the cost of a random walk sampling.

2) Global Proposal: In global proposal, we also have three proposals for the position, orientation, and layer index set X_s, X_o, and X_l, respectively.

Position Proposal: To make the new sample X_s^{k+1} jump far way from the local minimum, we sample the positions for all images independent of current state X_s^k.

Roughly speaking, all images in a good picture collage should not be overlapped, as shown in Fig. 1(c). To select initial positions of images without 1-D collage, we first divide the canvas C into a number of $N^* > N$ squares and randomly select a number of N centers s_i^c of squares without drawback. Then, we sample x_s from the distribution $\prod_i N(s_i; s_i^c, \sigma^2_{s_i^c})$, where $\sigma_{s_i^c}$ is 1/6 width of the square.

Oriental Proposal: The orientation x_o is sampled based on the prior of orientation:

$$q(x_o) \propto \prod_i N(o_i; 0, \sigma_o^2). \tag{25}$$

Layer Proposal: Layering is a unique property in the picture collage. Our layer proposal is a mixture of a random proposal and an $\alpha\beta$-swap search. The random proposal randomly selects a number of N layer indexes for x_l from $\{1, \ldots, N\}$ without drawback. The $\alpha\beta$-swap search is essentially the $\alpha\beta$-swap algorithm in graph cut optimization for a labeling problem on a graph. In our case, without changing positions and orientations, all images in the canvas construct a graph by connecting two overlaid images. The label of each node or image is the layer index. This $\alpha\beta$-swap search can be performed extremely fast if we only consider the salience maximization as the likelihood. Notice that our goal is to compute the MAP solution but not to truly sample the whole posterior. Our experimental results also show that this strategy works well for all results shown in the paper.

3) Pairwise Proposal: The acceptance rate of a global proposal is usually low compared with local proposal. In order to make the Markov chain have the ability to partially jump away from the local minimum, a pairwise proposal is designed for this goal. It can be viewed as a compromise between local proposal and global proposal. In each iteration, it only swaps the positions, orientations, or layer indexes of uniformly selected two different images. For example, to implement the kth iteration of a position pairwise proposal, we first uniformly sample the positions of an image pair (s_i^k, s_j^k), and construct the candidate set with a swap: $(s_i^{k+1} = s_j^k, s_j^{k+1} = s_i^k)$. Then let $s_i^{k+1} = s_j^k, s_j^{k+1} = s_i^k$ with the probability: $\min\{1, \pi(s_i^{k+1}, s_j^{k+1}, *)/\pi(s_i^k, s_j^k, *)\}$, and let $s_i^{k+1} = s_i^k, s_j^{k+1} = s_j^k$ otherwise. It is similar to implement the pairwise proposal of orientations or layer indexes. This proposal is in particular useful in the early and intermediate phases of the sampling.

4) Dynamic Weighting: The three weights v_l, v_g, and v_p in (23) represent our expectation on the frequencies of the local, global, and pairwise proposals being utilized. In other words, in each iteration of MCMC, the three weights are used to determine which proposal will be sampled to generate the parameter status. Practice shows that different proposals have different roles for the optimization. Therefore, we propose a dynamic weighting scheme to adaptively sample the proposals as the following. When the local proposal cannot improve the result in a longer time, the global and pairwise proposals should have larger probabilities to be utilized. So we set $v_l = \exp(-t^2/2\sigma_t^2)$, where t is the iteration number that the local proposal does not improve the result continuously, σ_t controls the probability that the local proposal is utilized, and is set as $6N$ so that three types of local proposals have a good chance to be utilized. The pairwise proposal also has the potential to partially jump away from the local minimum and find local details. So it is desirable to encourage the pairwise proposal to have a higher probability to be utilized than the global proposal. Therefore, we set $v_p = 3v_g = 3(1 - v_l)/4$.

C. Analysis on MCMC Optimization

Two examples from MCMC optimization are shown in Fig. 9. In these examples, all images are arranged automatically in the

(a) Automatic collage (b) Interactive collage

Fig. 10. Interactive picture collage. (a) Picture collage is created automatically, while the most important regions from each image are visible in a limited canvas. (b) Interactive optimization creates a very natural storyboard, with about 10–15 simple operations such as dragging and clicking. We can read the story at the first sight: a group of students visit Microsoft, and attend many talks, with other snapshots including sea, buildings, stores, and so on.

canvas, while all salient regions are shown without occlusions. The convergence condition in MCMC optimization is that the energy decreasing during a cycle is smaller than a threshold (0.1 in the paper). The quantitative analysis from the following experiments indicates that MCMC optimization can achieve a nice collage in 1–2 s for less than 30 images.

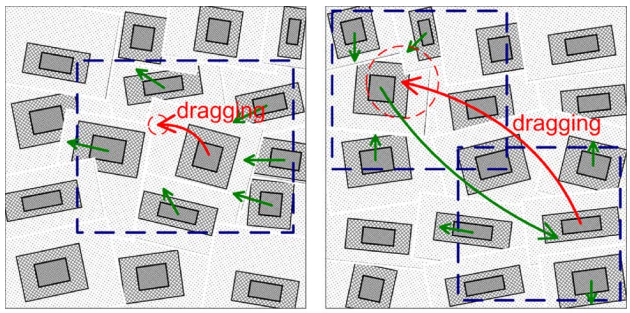
V. Interactive Optimization

The collaboration of quick initialization and MCMC optimization can achieve a nice collage for a large number of images very efficiently. However, in some cases, users want to interact so that they can specify their preferences during collage. For example, the collage in Fig. 10(a) is automatically generated from the pictures of a journey. The user wants to adjust the positions of images to make the group photo stand out as in Fig. 10(b). In the real application, it is impossible to assign the positions for hundreds of images. If the interaction can be integrated with automatic optimization, we can assign the positions of some images, while other images will be smartly arranged automatically. Can we successfully integrate the interaction and the automatic optimization? There are some key problems: how to automatically optimize all images after each operation on one image; how to speed up the optimization for real-time interaction, especially when the number of images increases; and so on. To solve these problems, the dragging operation is analyzed as an example in detail.

A. Dragging Operation

1) Formulation: When the user drags an image to a target position s_{iU}, a Gaussian distribution is preferred on the target position: $E_U(s_i, U) = -\beta_s \log(N(s_i; s_{iU}, \sigma_s^2))$. There are two kinds of dragging:

- Local dragging. When an image has been dragged only a small distance, it can be inferred that the user is fine tuning. In this case, a very small variance σ_s is assigned to prevent the following automatic optimization from changing the user's intention.



(a) (b)

Fig. 11. Dragging operation. (a) Local dragging and (b) long-distance dragging. The red arrow is dragging from users, and all green arrows denote images being automatically moved when this task has been completed. The terminal circle means the possible position from dragging, and the size corresponds to the variance of distribution. The blue rectangle with dashed lines is the regions where the visible salient rectangles must be re-computed.

- Long-distance dragging. In the case that the user wants to arrange one image in the left-top corner of the canvas, a large variance σ_s is assigned, which allows for further automatic optimization after dragging.

How does one discriminate between local dragging and long-distance dragging and assign the variance automatically? A polygon P_i surrounding the current image is computed through connecting the centers of the neighboring images. When this image is dragged outside the polygon P_i, it can be inferred that the user make a long-distance dragging. Otherwise, it can be inferred that the user make a local dragging.

2) Optimization After Dragging: Given these two kinds of dragging, can we automatically optimize all images following the operation.

- For local dragging, the local proposals will be run on the surrounding images, and these images will move aside along the green arrows in Fig. 11(a) under the control of the energy constraints.
- For long-distance dragging, the image nearest to the target position is selected to make a switch, which is shown as

(a) (b)

Fig. 12. Image summarization using picture collage. (a) is with 18 images from the search engine with keyword "bear". (b) is with eight photos collected for a child. Both collages are created automatically.

the long green arrow in Fig. 11(b). After this pairwise proposal, all local proposals are run on the surrounding images of the two switched images.

3) Speed Up Optimization: Efficiency is very critical for an interactive system. The computation of the visible salient rectangles for all images after each operation is time taken and is the bottleneck of the optimization algorithm. Therefore, we only update the visible salient rectangles of the neighboring images for each local proposal. For the dragging operation, the updated regions are approximately shown as the blue dashed rectangles in Fig. 11. Experiments indicate this strategy can speed up the interaction optimization.

B. More Operations

More operations are similarly defined as the dragging operation and have the similar mechanism. The functions of these operations are introduced briefly to demonstrate what the interactive collage can do.

1) Orientation Operation: Besides the position, users can adjust the orientation for each image. It is also formulated as a Gaussian distribution with the center at the given orientation and a small variance. Local proposals will follow the operation on the current image and their neighbors.

2) Layer Operation: Users can assign a new layer index o_{iU} for each image. All images with layer index between o_i and o_{iU} will increase or decrease by 1, and this will keep the original layer order. Only local layer proposals are operated on the changed images and their neighbors.

3) Image Zoom: Users can zoom in or out on selected images. After each operation, the attention rectangle must be re-computed, and the local proposals are operated on the current image and all surrounding neighbors.

4) Canvas Operation: Users can adjust the canvas size and scale. After each operation, all the positions s_i are changed with a ratio computed from the current and previous canvas. This operation will induce all local and global proposals on all images.

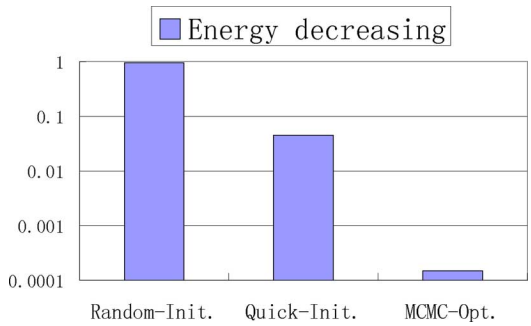


Fig. 13. Mean energy decreasing from the running of 100 times for four groups with 10, 50, 100, and 200 images. The vertical axis is the normalized energy, where the energy from random initialization is supposed to be 1. For 10, 50, and 100 images, the energies from quick initialization and MCMC optimization are around 5×10^{-2} and 2×10^{-4}. For 200 images, the energies change to 8×10^{-2} and 3×10^{-4}.

VI. Experiments

Experiments on picture collage run on a 2.8-GHz desktop PC with 1.0 G of memory. Suppose 1–2 s are the user's tolerance for each operation, and we call this time "lag-time". Our collage algorithm can achieve quick initialization for 300–500 images within lag-time boundaries. MCMC optimization converges within similar boundaries for less than 30 images. Interactive optimization runs for each operation on less than 50–70 images. Automatic quick initialization and MCMC optimization can progress on more than 500 images, while the intermediate results are outputted at an interval of 3 s. Users can select to pause or continue the optimization at any time to get an intermediate result, or interact with the system while interactional and automatic optimizations can cooperate to output a satisfactory collage.

The basic function of picture collage is to make a summarization of image search results and photograph collections. In Fig. 12, two examples are given from automatic picture collage. (a) is from the summarization of image search results with Google's image search, with the noisy images being discarded by hand. (b) is from family photography.

Fig. 14. Two collage examples with energy decreasing from the left to the right, with 22 images (top row) and 32 images (bottom row). It is observed that a lower energy corresponds to a nicer collage.

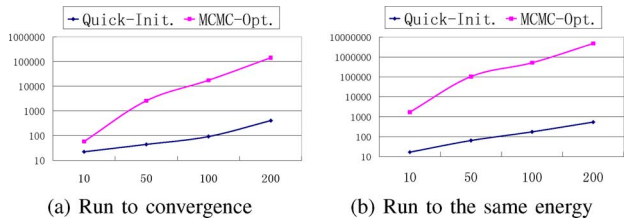
In the quantitative evaluation, the efficiency of quick initialization and MCMC optimization is tested. The parallel strategy is also introduced for a large number of images. The following experimental results are all obtained with a fixed set of parameters as mentioned above.

A. *Efficiency Evaluation*

The canvas has a 0.75 height/width ratio and its size is 60% of the total area of all input images. The salience rectangles are computed automatically beforehand.

1) Energy Decreasing: To demonstrate how optimizations reduce energy, four groups of images are collected with increasing numbers {10, 50, 100, 200}. Energy from three different algorithms is recorded: 1) random initialization; 2) quick initialization; and 3) MCMC optimization based on quick initialization (the convergence condition is the changed energy for each cycle that it is less than 0.1). For each group of images, each algorithm is repeated 100 times. The mean energies are computed for each group. To take off the influence from the image numbers, the energies are normalized using the energy from a random initialization for each group. The mean energies on groups are shown in Fig. 13. It is clear that the proposed quick initialization and MCMC optimization can reduce the energies efficiently, and the MCMC optimization can achieve lower energy than quick initialization by about 10^{-2} times. Two corresponding collage examples are shown in Fig. 14, and we observe that a lower energy corresponds to a nicer collage.

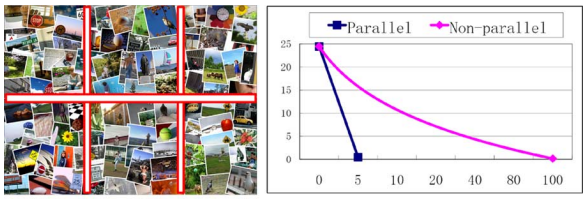
2) Time Taken versus Number of Images: For the above groups of images, the running times are also recorded for each algorithm. For each group, the mean time taken for 100 repetitions is shown in Fig. 15(a). Random initialization is not shown because its cost is very low. It is observed that quick initialization can finish in 0.2 s for 200 images. Based on quick initialization, MCMC optimization achieves lower energy (shown in Fig. 13) over a longer time.



(a) Run to convergence (b) Run to the same energy

Fig. 15. Time taken versus number of images. The vertical axis is the time in milliseconds, and the horizontal axis shows the number of images. (a) To converge for 10–200 images, quick initialization takes between 0.02–0.4 s, and MCMC optimization takes between 0.06–144 s starting from the results by quick initialization. (b) To achieve the same energy use, quick initialization costs 10^{-3} times the MCMC optimization.

Furthermore, we compare the time taken to get the same energy between quick initialization and MCMC optimization. Another four groups of images are repeated 100 times. The energy from the quick initialization is set as the convergence condition for the MCMC optimization which begins from the random initialization. As shown in Fig. 15(b), quick initialization costs 10^{-3} times the MCMC optimization to arrive the same energy.



(a) (b)

Fig. 16. Parallel collage. (a) Demonstrated sub-canvas for parallel collage. (b) Energy (vertical axis) versus time taken (horizontal axis with seconds) curve on 100 images. The starting energy is the random initialization, and parallel algorithm and non-parallel algorithm run on these images 100 times with the same convergence condition. It is observed that with parallel collage, the energy decreases quickly.

B. Parallel Collage on Sub-Canvas

A nice collage is efficiently outputted by quick initialization and MCMC optimization. However, when the number of images increases to hundreds, MCMC optimization still faces difficulties. Parallel collage proposes to group images and the canvas, because we observe that MCMC optimization based on quick initialization can run with a very high efficiency on a small number of images. If a large number of images ($N > 20$) is to be placed, the canvas is firstly partitioned into several sub-canvases. Then we perform collage inference in parallel on each sub-canvas. Afterwards, those sub-canvases are packed into the original large canvas, where two steps are followed to refine the collage: 1) run the local sampling on the boundary images and 2) run the local, global, and pairwise samplings for all the images on the whole canvas. An example is shown in Fig. 16(a), where the collage is divided into six sub-canvases. Two issues are addressed in the following.

1) Smart Grouping: Smart grouping is proposed to divide the canvas and images into several groups. Given the number of images N and the canvas width/height ratio $r = W/H$, how many groups should the images be divided into and how many images in each group? Experiments indicate that MCMC optimization has a very high efficiency for $n_g \in [5, 10]$ images. Then the number of groups should be $N_r \times N_c, N_r, N_c \in [N/10, N/5]$, where N_r and N_c are numbers of rows and columns. The resolve can be written as

$$\{N_r, N_c, n_g\}^* = \arg \min_{n_g \in [5,10], N_r * N_c \in [N/10, N/5]} (N_r * N_c * n_g - N) \quad (26)$$

where $N_r \approx N_c * r$, and $N_r * N_c * n_g \geq N$ promise that all images can be assigned a group. All images are grouped with a random order at each time, and there may be some groups with only $n_g - 1$ images.

2) Parallel versus Non-Parallel: To indicate how the parallel strategy improves efficiency, quick initialization followed by MCMC optimization will run by parallel strategy and non-parallel strategy, respectively, on 100 images, 100 times. With the same convergence condition, the time taken is recorded and the mean energy/time curves are shown in Fig. 16(b). Obviously, the parallel strategy decrease most quickly.

(a) (b)

Fig. 17. Results from Autocollage [12]: the cases that the images are blended with artifacts.

C. Comparison With Autocollage

We compare picture collage with Autocollage [12] which is an existing state of the art piece of software.[2] As introduced in Section I, Autocollage provides some specific traits, for example, images are ranked and selected automatically, and the images are blended in the final result. In most cases with less than 50 images, Autocollage can output a beautiful result with advanced blending technologies. Compared with Autocollage, our picture collage has four main advantages. 1) Our algorithm can output a quick initialization from 1-D collage as shown in Fig. 15, and it is much faster than Autocollage, especially when the number of images increases. For example, for the 32 images from Fig. 14, the trial version of Autocollage outputs the final result in about 28 s, while picture collage can output a quick initialization in much less than 1 s, as shown in Fig. 14(b). MCMC optimization can output the intermediate results every 3 s and will converge in about 10 s (including the time taken to generate the intermediate collages with high-quality images). We can stop this optimization at any time if we are satisfied with the current result. 2) We use the overlay style where the blending from Autocollage may bring artifacts in some cases. As shown in Fig. 17(a), the head of a woman from the left part of the collage is blended with another image; in Fig. 17(b), there are obvious artifacts on the boundaries of images. 3) Our algorithm integrates the interactive optimization well, and we can operate on each image if we are not satisfied with the automatically generated result. Fig. 10 provides an example where the group photo is dragged to the center of collage and zoomed in for a better view. Although Autocollage provides some user interactions, such as image rank, they do not provide the operation on each image, and further their interactions are not integrated in the optimization. 4) Our algorithm can work on hundreds of images and even on the canvas with arbitrary shapes as shown in the following subsection, and these traits are deficient in Autocollage.

D. Extension

Picture collage can be created on a canvas with arbitrary shapes as in Fig. 18, where the arbitrary shapes are uniformly processed in canvas shape constraint. Given a binary mask, edges are automatically extracted and fitted to several enclosed contours, which are expressed as the canvas polygon. This canvas polygon is similarly processed when we compute the energies and optimize with the MCMC algorithm. One issue is that quick initialization will decompose the spatial collage

[2]http://research.microsoft.com/en-us/um/cambridge/projects/autocollage/

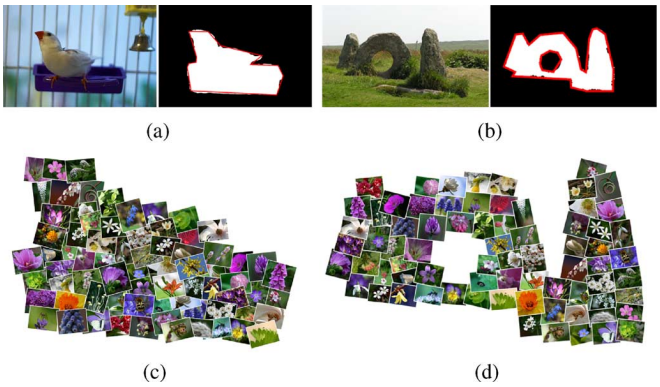


Fig. 18. Automatic collage on canvas with arbitrary shapes. The binary mask from the natural image is the input.

into a 1-D collage by grids-based layout. For an arbitrary shape, an approximate rectangle is computed with equal area, while quick initialization runs on this rectangle. Then a warp is operated on positions s_i to fit the arbitrary shape. Another issue is from several disconnected enclosed canvas contours. Images are divided into groups, and the sum of image areas for each group corresponds to the area of the enclosed contour. Then the collage will work on each enclosed contour, respectively.

E. User Study

To indicate that picture collage can create a natural, beautiful, and satisfactory collage with a high efficiency, a user study is conducted. Twenty volunteers are asked to create collages by themselves using ten groups of images with numbers 10–70; these users do not have any knowledge about collage and related experiences. We spend 1–2 min teaching them how to use picture collage. Then users will create their picture collage by themselves on all ten groups of images, and answer the following questions with 1 (definitely no) to 5 (definitely yes) after operating on each group. The averaged results are as follows.

- Do you think it is a natural and beautiful summarization? (4.2)
- Is the result from automatic collage satisfactory? (4.3)
- Do you think the interfaces for interactive optimization are useful? (4.2)
- Would you like to publish the picture collage on your web space? (4.6)

These results indicate that picture collage really creates a beautiful and satisfactory collage.

VII. Conclusion

In this paper, picture collage based on energy optimization is proposed to create a visual image summarization for a set of images, where different constraints from image salience, canvas, natural preference, and user's interaction are integrated in a CRF model. A two-step method including quick initialization and MCMC optimization is proposed to achieve high performance and efficiency. We also integrate interactive optimization to implement a semi-automatic collage. Future work will include: 1) a better salience analysis technique, e.g., incorporating more semantic object recognition; 2) incorporating high-level knowledge to create a storyboard on collage; and 3) more applications on photo summarization or browsing, especially on the Internet.

References

[1] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1026–1038, 2002.

[2] Li F.-F, R. Fergus, and P. Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. ICCV*, 2003, pp. 1134–1141.

[3] Y. Ma and H. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *ACM MultiMedia*. New York: ACM, 2003, pp. 374–381.

[4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[5] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proc. CVPR*, 2007.

[6] C. B. Atkins, "Adaptive photo collection page layout," in *Proc. ICIP*, 2004, pp. 2897–2900.

[7] J. Geigel and A. C. Loui, "Using genetic algorithms for album page layouts," *IEEE Multimedia*, vol. 10, no. 4, pp. 16–27, 2003.

[8] A. Girgensohn and P. Chiu, "Stained glass photo collages," in *Proc. 17th Annu. ACM Symp. User Interface Software and Technology*, 2004.

[9] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. F. Cohen, "Interactive digital photomontage," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 294–302, 2004.

[10] N. Jojic, B. J. Frey, and A. Kannan, "Epitomic analysis of appearance and shape," in *Proc. ICCV*, 2003, pp. 34–43.

[11] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake, "Digital tapestry," in *Proc. CVPR*, 2005, vol. 1, pp. 589–596.

[12] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake, "Autocollage," in *Proc. SIGGRAPH '06*, New York, 2006, pp. 847–852.

[13] V. J. Milenkovic, "Rotational polygon containment and minimum enclosure using only robust 2D constructions," *Comput. Geom. Theory Appl.*, vol. 13, no. 1, pp. 3–19, 1999.

[14] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-packing-based module placement," in *Proc. ICCAD '95*, Washington, DC, 1995, pp. 472–479.

[15] A. Santella, M. Agrawala, D. Decarlo, D. Salesin, and M. Cohen, "Gaze-based interaction for semi-automatic photo cropping," in *Proc. CHI*, 2006, pp. 771–780.

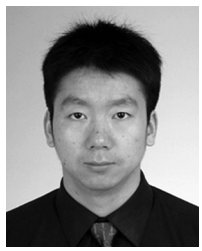
[16] R. Xiao, M. Li, and H. Zhang, "Robust multi-pose face detection in images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 31–41, 2004.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511–518.

[18] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[19] Z. Qin and J. Liu, "Multi-Point Metropolis Method with Application to Hybrid Monte Carlo", Tech. Rep., Harvard Univ., Dept. Statist., 2002, J. Comp. Phys., 172:827-40.

[20] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. New York: Springer, 2002.

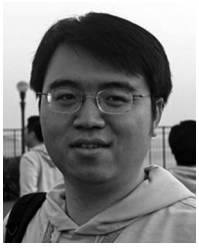


Tie Liu received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2007, respectively.

He is currently a staff researcher at the Analytics and Optimization Department, IBM China Research Lab, Beijing. His areas of interest include machine learning, pattern recognition, multimedia computing, and computer vision. He is also interested in data analysis and mining.

Jingdong Wang received the B.Sc. and M.Sc. degrees in automation from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 2007.

He is currently an associate researcher at the Media Computing Group, Microsoft Research Asia, Beijing. His areas of interest include machine learning, pattern recognition, multimedia computing, and computer vision. In particular, he has worked on kernel methods, semi-supervised learning, data clustering, image segmentation, and image and video presentation, management, and search.



Jian Sun received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1997, 2000, and 2003, respectively.

Then, he joined Microsoft Research Asia, Beijing, China, in July 2003. His current two major research interests are interactive compute vision (user interface + vision) and internet compute vision (large image collection + vision). He is also interested in stereo matching and computational photography.

Nanning Zheng (SM'93–F'06) received the B.S. degree from the Department of Electrical Engineering and the M.S. degree in information and control engineering from Xi'an Jiaotong University, Xi'an, China, in 1975 and 1981, respectively, and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985.

He jointed Xi'an Jiaotong University in 1975, and he is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems.

Dr. Zheng became a member of the Chinese Academy of Engineering in 1999, and he is the Chinese Representative on the Governing Board of the International Association for Pattern Recognition. He also serves as an executive deputy editor of the Chinese Science Bulletin.

Xiaoou Tang (S'93–M'96–SM'02–F'09) received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990 and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996.

He is a Professor in the Department of Information Engineering at the Chinese University of Hong Kong. He worked as the group manager of the Visual Computing Group at the Microsoft Research Asia, Beijing, China, from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing.

Dr. Tang has received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. He is a program chair of the IEEE International Conference on Computer Vision (ICCV) 2009 and an associate editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI) and *International Journal of Computer Vision* (IJCV).

Heung-Yeung Shum (F'06) received the Ph.D. degree in robotics from the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, in 1996.

He worked as a researcher for three years in the Vision Technology Group at Microsoft Research, Redmond, WA. In 1999, he moved to Microsoft Research Asia, Beijing, China, where his tenure began as a Research Manager and subsequently moved up to Assistant Managing Director, Managing Director of Microsoft Research Asia, Distinguished Engineer, and Corporate Vice President. His research interests include computer vision, computer graphics, human computer interaction, pattern recognition, statistical learning, and robotics.

Dr. Shum is the General Co-Chair of the Ninth International Conference on Computer Vision (ICCV) 2003 and a Program Chair of the International Conference of Computer Vision (ICCV) 2007. He is a Fellow of ACM.