Computational Visual Media
https://doi.org/10.1007/s41095-019-0130-7 Vol. 5, No. 1, March 2019, 45–58

Research Article

Discernible image mosaic with edge-aware adaptive tiles

Pengfei Xu[1], **Jianqiang Ding**[1], **Hao Zhang**[2], **and Hui Huang**[1] (✉)

Abstract We present a novel method to produce *discernible* image mosaics, with relatively large image tiles replaced by images drawn from a database, to resemble a target image. Compared to existing works on image mosaics, the novelty of our method is two-fold. Firstly, believing that the presence of visual edges in the final image mosaic strongly supports image perception, we develop an *edge-aware* photo retrieval scheme which emphasizes the preservation of visual edges in the target image. Secondly, unlike most previous works which apply a pre-determined partition to an input image, our image mosaics are composed of *adaptive* tiles, whose sizes are determined based on the available images in the database and the objective of maximizing resemblance to the target image. We show discernible image mosaics obtained by our method, using image collections of only moderate size. To evaluate our method, we conducted a user study to validate that the image mosaics generated present both globally and locally appropriate visual impressions to the human observers. Visual comparisons with existing techniques demonstrate the superiority of our method in terms of mosaic quality and perceptibility.

Keywords image mosaic; image retrieval; image synthesis

1 Introduction

An image mosaic or photographic mosaic [1, 2] is a picture (usually a photograph) that is divided into usually uniformly sized tiles, each of which is replaced by another photo, so that the entire mosaic resembles a target photo. As an art form, image mosaics have widely appeared in advertising, decoration, and entertainment. Ideally, results from such a "pictures in a picture" composition paradigm should provide both global and local visual impressions. Globally, when viewed afar or with purposely blurred vision, the mosaic should resemble the target photo in color and texture. At the same time, close examination should easily reveal the content of each component photo.

At one extreme, producing an image mosaic is trivial if the tiles are sufficiently small, e.g., the size of a pixel. Spatial integration within the human eye leads to perception of each small photo as a singly colored tile, leading to the best approximation of the target photo globally. However, the contents of the small photos are no longer recognizable. A compelling question is how to attain both global resemblance and local recognizability by use of *larger* image tiles in a mosaic, as shown in Fig. 2. The main challenge is that with larger tiles, close resemblance between the small photos and the target photo is harder to achieve and their visual differences are more apparent.

In this paper, we present a novel method to produce *discernible* image mosaics which resemble a target photo, using relatively large image tiles replaced by photos drawn from a database. Compared to existing works on image mosaics [3–8], the novelty of our method is two-fold:

- Firstly, we recognize the sensitivity of human perception to edge structures in images and develop a photo search mechanism that is *edge-aware*. Since detectable visual differences between the small photos and the target photo are inevitable, we elect to emphasize preservation of visual edges in the target photo over overall resemblance of texture information.

1 College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China. E-mail: P. Xu, xupengfei.cg@gmail.com; J. Ding, dingjianqiang0x@gmail.com; H. Huang, hhzhiyan@gmail.com (✉).

2 School of Computing Science, Simon Fraser University, Vancouver, BC, Canada. E-mail: hao.r.zhang@gmail.com.

Manuscript received: 2018-12-09; accepted: 2019-01-20

- Secondly, most previous mosaic works apply a predetermined partition to the input image, while our image mosaics are composed of *adaptive* tiles, whose sizes are determined based on the available photos in the database and the objective of maximizing resemblance to the target photo.

To realize edge-aware photo retrieval, we adopt a weighted L_2 norm to measure the similarity between two images. The weighting scheme depends on edge features present in the query image so that edge similarities are emphasized. Furthermore, to reduce the need for exact color matching, we introduce a color offset term when computing the image similarity distance.

Adaptive tiles are determined incrementally, using a scan order across the input image. Joint tile size optimization and maximization of image resemblance is carried out via dynamic programming. Compared to the use of fixed partitions for image mosaics, our adaptive partition scheme is able to exploit the full potential of photos in the database to improve the quality of the final mosaic result.

The database for our mosaic generation consists of photos "in the wild", such as those from an online image search or existing image repositories. To keep the search cost to a reasonable level, we work with photo datasets of moderate size (here, 180,000 images). With our edge-aware retrieval and database-adaptive image partitioning, our method attempts to make the best out of this limited set of photos.

Figure 1 shows discernible image mosaics obtained by our method, which exhibit both global resemblance to the target photo, and local recognizability of the photo tiles. There is still a gap in quality and artistry compared to an artist's creation such as that shown in Fig. 2. However, it is worth noting that the small photos therein were hand-crafted by the artist and did not come from a generic photo collection.

To evaluate our method, we conducted a user study to validate that the image mosaics generated present both globally and locally appropriate visual impressions to the human observers. Visual comparisons with existing techniques demonstrate the superiority of our method in terms of mosaic quality and perceptibility.

2 Related work

2.1 Traditional mosaics

Mosaicing is a historical art form, producing a picture or pattern composed of a set of small colored or textured tiles [9]. Nowadays, people often create mosaic images in digital form using computational approaches. A mosaic image can be created by segmenting an ordinary image into small regions by Voronoi tessellation [10, 11], or polygon tessellation [12], and then filling these closely neighboring irregular regions with constant colors or textures. Other works generate mosaic images from disconnected regular tiles. Their objective is to arrange a set of tiles with identical shapes to represent the content of an image in 2D, or a surface in 3D [13]. The positions and orientations of the tiles can be determined by computing a centroidal Voronoi diagram [14], minimizing an energy function with graph-cut [15], or constructing a gradient vector flow field [16]. The color or texture of each tile is determined by the original image. The mosaic images generated by all these techniques can be considered to be a stylistic representation of an input image.

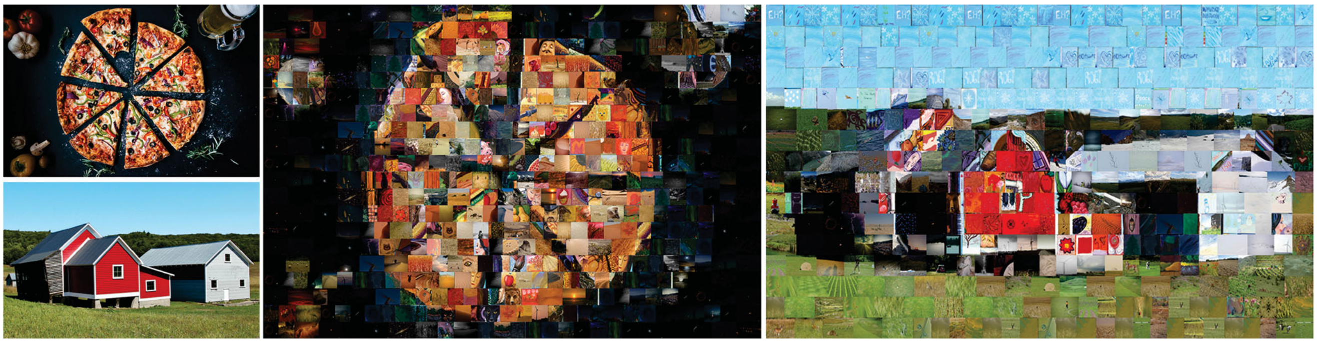

Fig. 1 *Discernible* image mosaics generated by our method (center, right). The replacement photographs in the mosaic tiles remain recognizable while together they resemble the target photographs (left). Two unique features of our method are the use of *adaptive* mosaic tiles, whose sizes vary, and *edge preservation*, e.g., see outlines of the cabin.

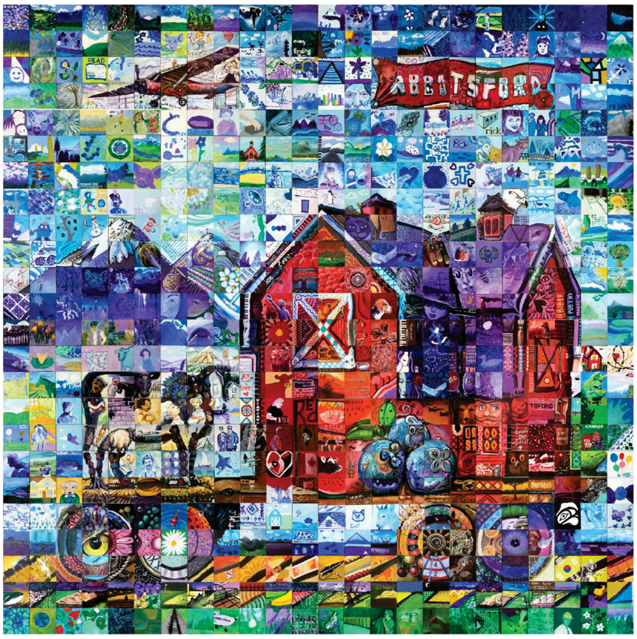

Fig. 2 A discernible image mosaic created by an artist. The presence of visual edges (of the house, cow, airplane, etc.) strongly supports perception of the objects while their textures are more artistic, showing clearly discernible contents, than realistic. Note that the small photos were hand-crafted by the artist, and were not retrieved from a generic photo dataset.

2.2 Image mosaics

Image mosaics or photomosaics [2] are a variation of the traditional mosaic art form, and are also composed of a set of tiles. Instead of constant colors or textures, the tiles in image mosaics are themselves images. The tiles are not created by texture synthesis [17, 18], but retrieved from a database. The appearances of these images resemble the local content of the target image, and together represent the content of the whole target image. Creation of an image mosaic typically occurs in two steps, firstly determining the tile set, and then finding a replacement image for each tile. Di Blasi et al. [4] introduced a grid-based image descriptor and a tree data structure to accelerate the image replacement step. Orchard and Kaplan [5] allowed each target image tile to be replaced by a local part of an image in the database, using FFT to reduce the computation in evaluating matches between local parts of images and the tiles of the target image. Barnes et al. [19] utilized a *PatchTable* data structure to reduce the query time for image patches. Pavić et al. [6] adopted a polynomial descriptor to approximate the content of images. When replacing the tiles of the target image, descriptors with different degrees are adaptively determined based on a feature/non-feature classification. After the initial replacement, a merging step is used to increase tile size in non-feature areas. Zhang et al. [8] also adopted a tile merging step guided by region entropy to reduce the number of tiles. This group of techniques is closely relevant to our work. While their tile set is typically determined by regular partitioning of the target image, we adopt an adaptive database approach to determine the tile set, allowing us to fully exploit the potential of the database to ensure tile replacement quality. Like Refs. [6, 8], we aim to reduce the number of tiles, or equivalently, increase the tile size. Unlike these methods which focus on increasing the tile size in non-feature areas, our edge-aware image retrieval approach achieves better feature resemblance, enabling large replacement tiles in feature areas.

2.3 Content assembly with image objects

Assembling global content with small elements has long been a topic of interest. Instead of using general images, many works utilize well-defined image objects to form the global content [3, 20–22]. Kim and Pellacini introduced *Jigsaw Image Mosaics* [3], which use segmented image objects to fill the regions in the target image. The image objects are packed closely, and their boundaries together approximate the region boundaries in the target image. Di Blasi et al. [20] also achieved similar results, but with reduced computation. Huang et al. [22] presented an approach for creating Arcimboldo-like collages, in which a segmented image object is usually used to replace an entire region in the target image. Kwan et al. [21] introduced a pyramid of arc-length descriptors to improve the packing layout of image objects when filling regions. The boundaries of the objects also better resemble the region boundaries. Reinert et al. [23] designed an interactive system for manipulating the layout of image elements inside regions. The layout of the elements is automatically beautified after the placement of example elements by the user. Zou et al. [24] introduced an efficient algorithm to create legible compact calligrams, meaningful shapes composed of deformed characters. These works use well-defined objects to assemble global content, with deformation if necessary. In contrast, our work takes general images as elements to form global content, and uses the edge features of the images in the database to approximate the edge features in the target image. Hu et al. [25] introduced a novel hierarchical representation of images called *PatchNets* to enable fast creation of new images by replacing image objects. Zhang et al. [26] presented *PlenoPatch* which enables image object manipulation in a given image.

2.4 Image collages

Image mosaics can also be considered to be a collage of images. Existing works for creating image collages often focus on the aesthetic aspect, i.e., creating a pleasant layout of images. They do not use the image to form meaningful global structure or content. Rother et al. [27] proposed a labeling optimization framework for creating visually appealing collages from a set of images. Yu et al. [28] solved this problem using a power-diagram-based circle packing algorithm. Puzzle-like collages [29] exploit the boundaries of objects or regions of interest in the images, and can create a more compact layout. Since these works do not try to form a global content, they have much more freedom when placing the images compared to our work. In terms of delivering multiple pieces of information, our work is also related to hybrid images [30], camouflage images [31], and hidden images [32], in all of which the generated image is typically composed of a small number of images.

3 Method

An image mosaic is often obtained by first diving a target image into a set of tiles and then replacing each tile with a database image. In comparison to existing image mosaic work [3, 5, 7, 20], our method adopts an edge-aware retrieval procedure for tile replacement, making each replacement image capable of recovering the important edge features of the target image. In addition, our target image partitioning strategy adaptively determines the tiles from the database, reducing the matching error of the replacement image for each tile, thus improving the quality of the final image mosaic.

3.1 Edge-aware image retrieval

In this section, we describe our edge-aware image retrieval procedure. For now we assume that the target image is already divided into a set of tiles. As shown in Fig. 3, our system adopts a grid-based descriptor [4, 5, 8] to encode each image in the database or a tile in the target image: we transform the image or tile into a regular grid and concatenate the mean color values of all cells to obtain a vector. To keep enough information about the image, we use a dense grid [5], e.g., a 24 × 24 grid for a square tile, when computing the description vector. Thus, the description vector $\boldsymbol{v}$ has the form $(\boldsymbol{c}_1^{\mathrm{T}}, \ldots, \boldsymbol{c}_k^{\mathrm{T}})^{\mathrm{T}}$, where $\boldsymbol{c}_i$ is the RGB color of the i-th cell, and k is the total number of cells. Before computing the description vector, we apply edge-preserving smoothing [33] to the image to remove details, since they may make the description vector noisy. To straightforwardly measure the similarity of two images, we may compute the L_2 distance between the corresponding vectors. Given an image database, we construct a K-D tree structure from the description vectors of the images. Given a tile in the target image, we compute its description vector and efficiently retrieve its replacement image by use of the K-D tree structure [34].

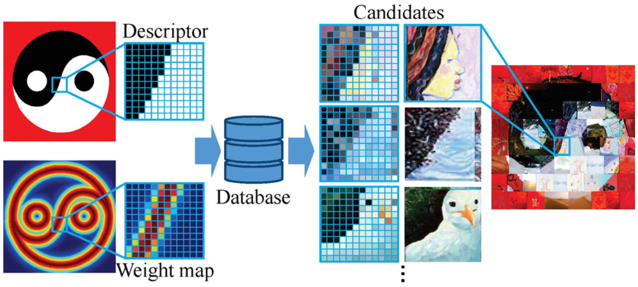

Fig. 3 Edge-aware image retrieval procedure.

The above image similarity measurement using L_2 norm treats each grid cell equally and does not emphasize any salient features. We recognize that edge features play an important role in defining the content of an image and those in tiles should resemble those in the target image (see Fig. 2). Unlike existing methods [6] which reduce tile size to better match edge features, we aim to produce resembling edge features using relative large image tiles. To emphasize edge similarity in the replacement image, we initially attempted concatenating an edge feature vector, the HOG descriptor [35], in the description vector of the image. The results were not promising, since the relative contributions of edge and texture similarity were difficult to control, and may even compete to such an extent that the retrieved image neither resembles the target's edges nor textures. We observed that edge features are actually formed by textures: two neighboring regions with different textures form a clear edge feature. Therefore, matching of edge features can be realized by matching textures. Based on this observation, we adopted a weighted L_2 norm to measure image similarity, and achieve edge-aware image retrieval. Given a tile in the target image, we first extract its salient edge features [36]. The areas near edge features should contribute more when measuring image similarity. We thus generate a weight map using Gaussian diffusion based on the edges, then the image similarity error between a tile T and a database image I can be computed as

$$E(T, I) = \| \boldsymbol{W}_T(\boldsymbol{v}_T - \boldsymbol{v}_I) \|_2 \tag{1}$$

where $\boldsymbol{W}_T$ is a diagonal weight matrix. An entry w in $\boldsymbol{W}_T$ corresponds to a grid cell and its value is defined as $w = 1 + \lambda \mathrm{e}^{-d^2/\bar{d}^2}$. d is the minimal distance between the grid cell and the edge feature. $\bar{d}$ controls the scope of the emphasized cells, and we set it to $\frac{1}{4}$ of the tile height. λ controls the degree of edge emphasis and is 10 in our implementation. We adopted the flann library [34] and modified the L_2 distance function to compute Eq. (1). $\boldsymbol{W}_T$ is considered to be a parameter, and is input to the distance function for each retrieval.

The distance defined by Eq. (1) emphasizes similarity in feature areas. This similarity measure heavily depends on exact texture matching. A replacement image with greater texture similarity but lower edge resemblance may be preferred in this setting. We observed from the image mosaic created by an artist in Fig. 2 that people are sensitive to edge structures and can tolerate a certain range of variations in texture. This inspires us to believe that, without loss of emphasis in edge matching, the similarity measure between the original tile and the replacement image could be relaxed from exact texture matching. We thus modify Eq. (1) to

$$E(T, I) = \min_{\Delta \boldsymbol{v}} \| \boldsymbol{W}_T(\boldsymbol{v}_T - \boldsymbol{v}_I + \Delta \boldsymbol{v}) \|_2 \tag{2}$$

where $\Delta \boldsymbol{v}$ is an offset vector that is used for relaxation of exact texture matching. To avoid evident texture mismatches in the retrieved image, we constrain each color offset $\Delta \boldsymbol{c}_i$ in $\Delta \boldsymbol{v} = (\Delta \boldsymbol{c}_1^{\mathrm{T}}, \ldots, \Delta \boldsymbol{c}_k^{\mathrm{T}})$ to be in a certain range: each RGB color channel in $\Delta \boldsymbol{c}_i$ is in the range $[-\Delta d, \Delta d]$. By introducing this color offset vector, the color histograms of the target image and the retrieved image need not exactly match. Since the color offsets $\Delta \boldsymbol{c}_i$ in the cells can be different, the transformation between these two histograms is composed of multiple independent color translations. We realize Eq. (2) by further modifying the distance function in the flann library, as follows:

$$\begin{aligned} D(\boldsymbol{u}, \boldsymbol{v}) &= \sqrt{\textstyle\sum_i (w_i d(u_i, v_i))^2} \\ d(u_i, v_i) &= \max(0, |u_i - v_i| - \Delta d) \end{aligned} \tag{3}$$

In our implementation, Δd is set as 15 for color channels in the range $[0, 255]$. The above definition is not actually a metric, since it does not fulfil the triangle inequality. However, this does not matter for the K-D tree search procedure.

Image shape adaptive description vector. Our system keeps the original shapes of the database images when assembling the target image. For images of different shapes, our grid-based descriptors also have different sizes, so the range of the matching error is proportional to the image sizes. In our current implementation, we compute the description vector of a square image based on a 24×24 grid, and an image with aspect ratio $W/H = 4/3$ based on a 32×24 grid. Grids for images with other aspect ratios can be derived similarly.

Partial image retrieval with weighted L_2 distance. The weighted L_2 norm can also be used for partial image retrieval [5]. Given a tile, we can set the weights as non-zero values in areas of interest, and zero elsewhere. Then the distance computed by Eqs. (1) and (2) is not affected by entries with weight zero, precluding retrieval outside areas of interest. We will describe how we benefit from this partial image retrieval in Section 3.2.

3.2 Database-adaptive target image partition

Existing image mosaic techniques [5–7] treat target image partitioning as a preprocessing step before tile replacement. Their partitions are regular grids, or guided by the content of the target image. Such schemes ignore the content of the database, making the generated tiles vulnerable to low quality image replacements. We adopt a database-adaptive target image partitioning scheme. Partitioning is determined based on the available images in the database, so the generated tiles are more likely to contain desirable replacement images.

We consider the partitioning process to be an optimization problem, involving selection of rectangular tiles. Given a target image, we need to select a set of tiles $\mathcal{T}$ from the target image Q, such that (i) the tiles together cover the target image, and (ii) the sum of retrieval errors over all tiles is minimized. This problem can be formulated as

$$\mathcal{T}^* = \operatorname{argmin}_{\mathcal{T}} \left(\sum_{T\in\mathcal{T}} E(T) \right),$$
$$\text{such that} \quad \bigcup_{T\in\mathcal{T}} T = Q \qquad (4)$$

where $E(T) = \min_{I\in\mathcal{I}} E(T,I)$, and $\mathcal{I}$ is the image dataset. Directly solving the above optimization problem is difficult. There is no explicit relation between the overall retrieval errors of different tile sets, so an analytical approach is not applicable. In addition, the number of tile sets which fulfil the constraints is huge, so it is not practical to adopt an exhaustive approach.

To obtain a feasible solution, we need to narrow down the search space. Existing works constrain tile sets to have grid-like layout with differing numbers of tiles. This constraint is too strong, leading to a small number of usable tile sets. As the images in the database have different aspect ratios, it is natural to modify the grid layout by allowing the tiles to have different widths, resulting in tile sets with brick wall layout. The number of such tile sets is huge. Consider a row of n tiles. if each tile has m possible widths, the number of combinations is $O(m^n)$. The huge number of applicable tile sets ensures that the best one can approximate the target image well.

A useful property of such tile sets is that the tiles have a clear linear order, which means the tiles can be determined incrementally. We thus use a dynamic programming approach to obtain the optimal tile set. As shown in Fig. 4, we first partition the target image into a number of rows. For each row, we select the tiles from left to right one by one. Since the images in the database have different aspect ratios, each time when selecting a new tile, there are several options for the tile shape, leading to several branches for the tile sets. For each tile option, we use the accumulated matching error of all selected tiles in this branch to update the minimal error record at the rightmost location of this tile. The accumulated error can be computed by an addition operation on all matching errors of the selected tiles, since the image shape adaptive descriptor ensures that the range of the matching error of each tile is proportional to the size of the tile. We continue the selection until all possible branches reach the rightmost location of the row. By using dynamic programming, we can efficiently obtain the optimal tile set with minimal matching error for the row. Repeating this procedure for each row generates a complete tile set, used for creating the image mosaic.

Relaxation of the tile shape. In the above image partition algorithm, the shapes of the tiles are determined by the shapes of the database images. We can introduce tiles with other shapes by allowing partial image retrieval. This can be achieved by using the weighted L_2 norm (Section 3.1). This relaxation increases the number of tile candidates for selection, and therefore may further improve the quality of the final image mosaic. In our implementation, we constrained the size of partial images to be at least 80% of the original images, allowing image contents to be preserved in the partial images.

Repetition control of the replacement images. For target images with repeated textures, the generated image mosaic may contain duplicated replacement images. To avoid apparent repetition artifacts, we record the replacement images used during the partitioning procedure, and constrain multiple successive replacement images to be different. In our implementation, we consider 10 such images to avoid repetition artifacts.

Generalization of tile layout. Although we introduced our partition procedure using brick–wall like tile layout, our method can easily be generalized to other layouts, as long as the tiles have a clear linear order. For example, our method may adopt a vertical tile selection procedure, based on columns rather than rows. It is also possible to combine horizontal

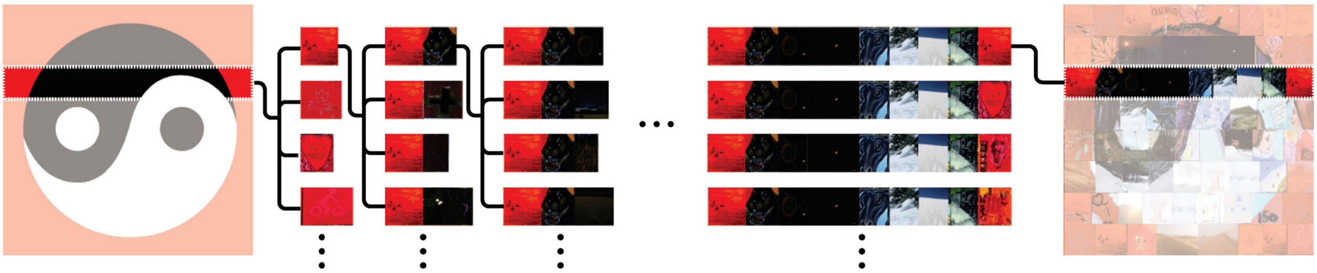

Fig. 4 For each row of the target image, we obtain the optimal tile set using dynamic programming.

and vertical tile selection procedures to create more interesting tile layouts. Here, the target image needs to be first manually partitioned into horizontal and vertical strips, using our algorithm to determine the tile images in each strip. It is also possible to adaptively determine the combination of horizontal and vertical strips according to image content. The algorithm for determining the image strips is beyond the scope of this paper, and is potential future work.

4 Evaluation and discussion

We tested our method with a broad class of images. Figures 1 and 11 show some results using our method. Figures 6 and 9 show image mosaics produced by our method with different numbers of rows. The edges of the target images are well-preserved, and the global contents are faithfully recovered, even with a small number of rows (e.g., using 5 rows for the tai chi diagram and 12 rows for the eagle and flamingo image). The content of each tile can also be easily recognized. All image mosaics were created using the same image database, with about 180,000 images. Most images were taken from the database used in Ref. [37]; some were downloaded from free image websites, such as Flickr, under a Creative Commons license. All images were added to the database unmodified. With this database, our method produces image mosaics reasonably quickly, in general, taking less than 3 min to prepare a single row of an image mosaic. Since the computation of each row is parallelizable, the total computation time for a complete image mosaic is less than 5 min. The computation time was recorded on a PC with a 2.3 GHz Xeon CPU and 64 GB RAM.

We have evaluated our method thoroughly. We evaluated the effectiveness of our database-adaptive target image partitioning scheme and edge-aware retrieval procedure separately. We compared our results with the state of art to show the advantages of the proposed method. We also investigated how the database might affect our method. Finally, we conducted a user study to evaluate our method from the user's point of view.

4.1 Evaluation of algorithm

To investigate the effectiveness of the two components of our algorithm, we generated image mosaics as follows: (a) keep edge-aware image retrieval, replace database-adaptive image partitioning with regular partitioning, (b) keep database-adaptive image partitioning, replace edge-aware image retrieval with retrieval using simple L_2 norm, (c) use both edge-aware image retrieval and database-adaptive image partitioning. Figure 5 shows image mosaics generated with these three approaches. In the first case, since the tile layout is constrained, the results cannot exploit the full potential of the database. In the second case, since the retrieval procedure is not edge-aware, edge preservation is affected. In the third case, the results achieve the best overall resemblance of the edge features to those in the target images. This indicates the importance of using both components of our algorithm to generating good image mosaics.

4.2 Comparison with other methods

We compared our method with other representative techniques that generate image mosaics with similar styles to ours. These techniques use grid-based image descriptors [4, 5, 8] or polynomial image descriptors [6] for image retrieval, and treat image

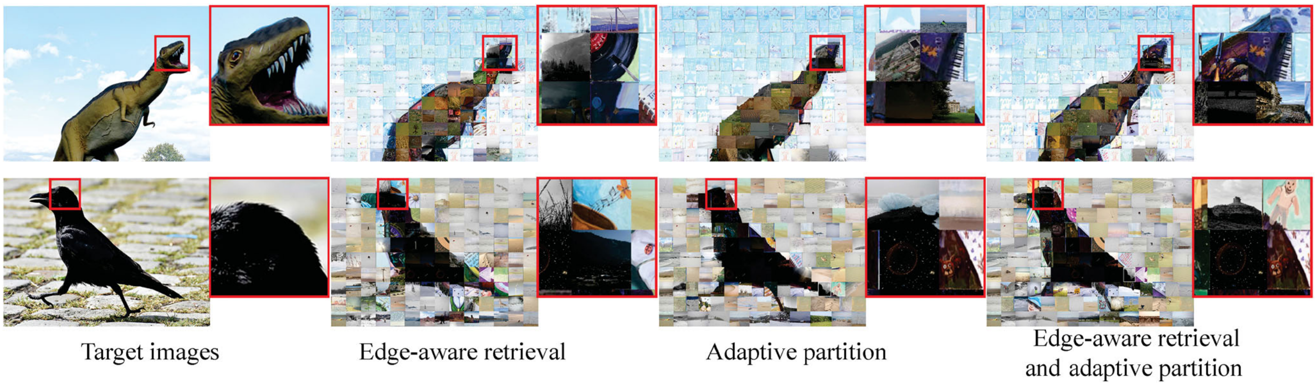

Fig. 5 Comparison between our method and simplified versions, showing the importance of both components of our algorithm for generating good image mosaics.

partitioning as a preprocess before tile replacement. We also compared with Foto-Mosaic-Edda (FME) [38], well-known commercial software from Rapid-Mosaic, whose algorithm is private. Techniques that produce traditional mosaics [10, 15], or use segmented image objects [3, 21, 22] as input were not considered in this comparison. The comparison focused on target image partitioning and replacement image retrieval. Other processes such as tile merging [6, 8] were not considered. All compared techniques used the same image database described above.

Figures 6 and 7 show the results. Figure 6 shows that the results generated using a sparse grid descriptor [4, 8] (3 × 3 grid, 27-dimension vector) only poorly preserve edges. This is understandable since the sparse grid descriptor loses edge information when encoding the image. Results generated using a dense grid descriptor [5] (24 × 24 grid, 1728-dimension vector) are slightly better, as the dense grid descriptor keeps more information from the image. However, because it does not emphasise edges, its performance is also unsatisfactory. The polynomial descriptor [6] has a similar problem to the sparse grid descriptor. When we increase the order of the polynomial descriptor to 1728 dimensions, there is no significant improvement. The results generated by the commercial software Foto-Mosaic-Edda are also deficient in preserving edges, implying that it is not designed to do so. It is worth noting that, even with fewer tiles, our method is able to create better image mosaics than existing methods. Figure 7 shows further comparisons. Due to the poor results from the sparse descriptors, they are not included in this comparison. These examples show that image mosaics generated by our method have the best edge preservation of any methods considered.

4.3 Effect of database

As for other image mosaic methods, our method is affected by the quality of the database. In general, a larger database is preferable since it can provide more candidates for tile replacement. To investigate how our method is affected by the database, we generated image mosaics with databases of different sizes. We prepared four datasets with 180,000, 90,000, 50,000, and 20,000 images, by gradually removing images. Figure 8 shows image mosaics created using these datasets: as the size of the dataset decreases, the quality of texture matching and edge matching in the produced image mosaic also decreases. However, even for the smallest database, our method can still generate reasonable image mosaics. This indicates that our method has the ability to exploit the full potential of the database, and so is more tolerant to low-quality databases. On the other hand, it is also noticeable that even for the image mosaic created from the largest dataset, there are still some artifacts. Indeed, artifacts are inevitable for a given database, since a limited number of database images cannot cover all the variation in target image details.

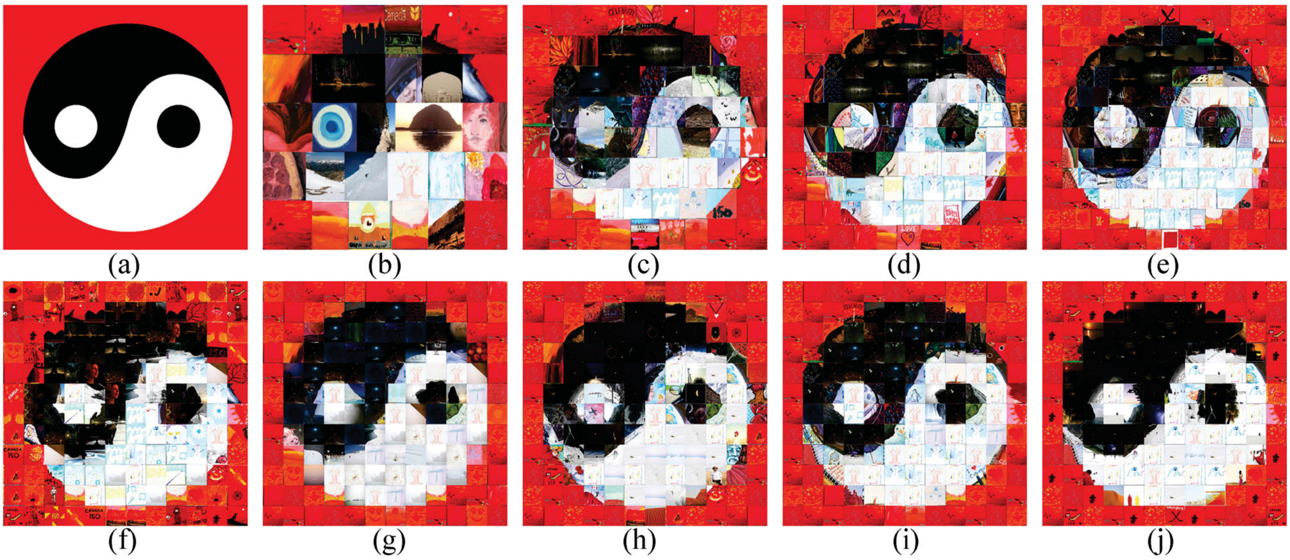

Fig. 6 Image mosaics for the tai chi diagram. (a) Target image. (b–e) Our results with 5, 8, 10, and 12 rows of tiles. (f) Result created by Foto-Mosaik-Edda. (g) Result with dense polynomial descriptor. (h) Result with sparse polynomial descriptor. (i) Result with dense grid descriptor. (j) Result with sparse grid descriptor. Images (f–j) have 12 rows of tiles.

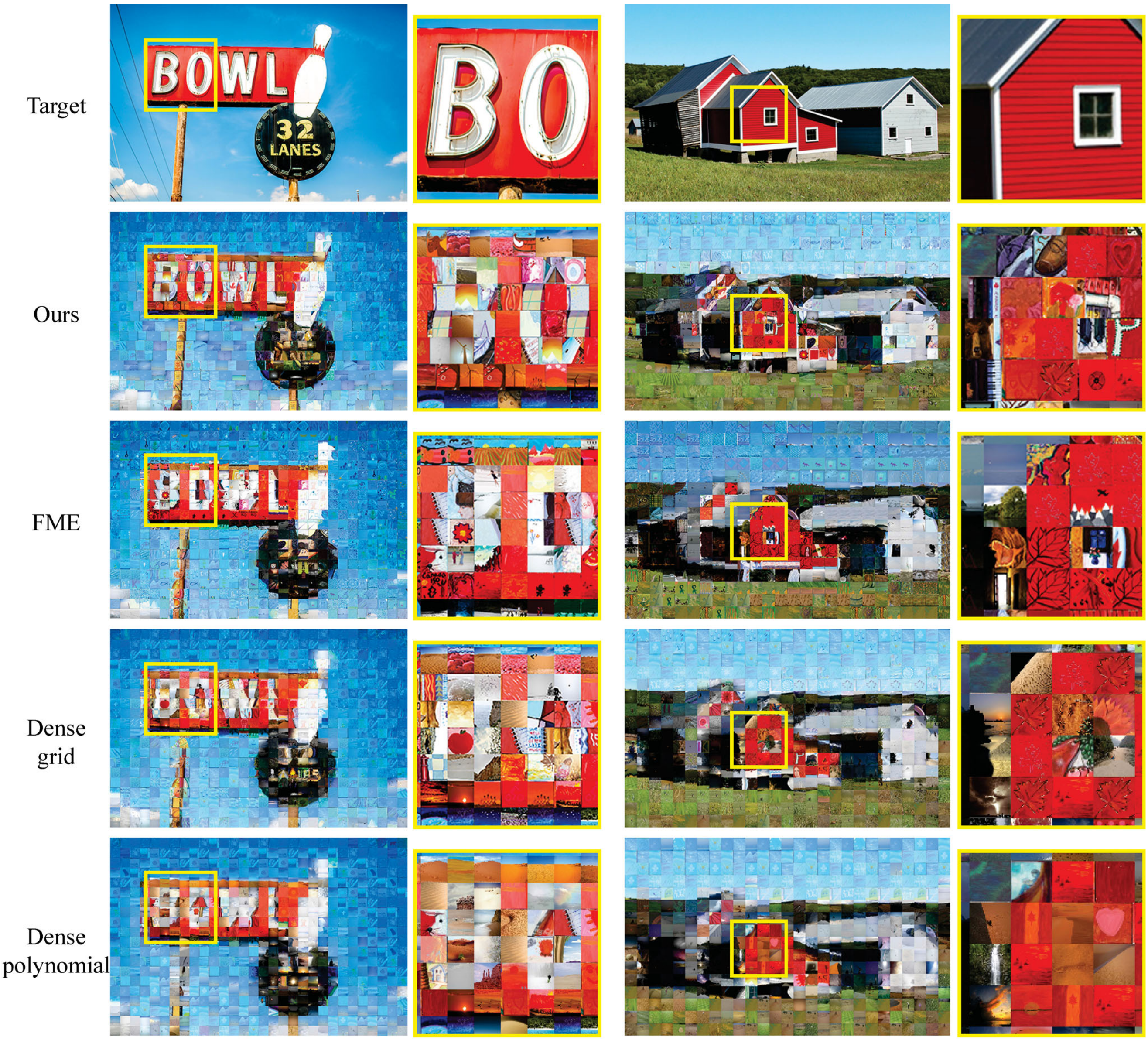

Fig. 7 Comparison with existing methods. Our method results in best edge preservation.

However, it is expected that artifacts will finally become invisible if sufficient images are included in the database.

4.4 User study

We conducted a user study to investigate (i) people's preferences regarding image mosaics generated by different approaches, and (ii) whether people can perceive the global and local visual content from an image mosaic with given size. 19 college students participated in this study.

Before the study, we prepared 5 sets of image mosaics. The target images of these 5 sets were: the cabin image in Fig. 1, the flamingo image in Fig. 9, the tyrannosaur image in Fig. 5, the pyramid image in Fig. 11, and the spade image in Fig. 8. Each set contained 3 image mosaics, which were generated using the following approaches: **A1**. Regular partitioning with image retrieval based on dense grid descriptors [5]. **A2**. Regular partitioning with image retrieval based on dense polynomial descriptors [6]. **A3**. Adaptive partitioning with edge-aware image retrieval.

During the study, we displayed each set of image mosaics on a monitor, with image mosaic height of about 13 cm. The relative positions of the image mosaics in a set were random. On viewing each set of image mosaics, the participants were requested to rate them from 1 to 5, where 1 meant completely

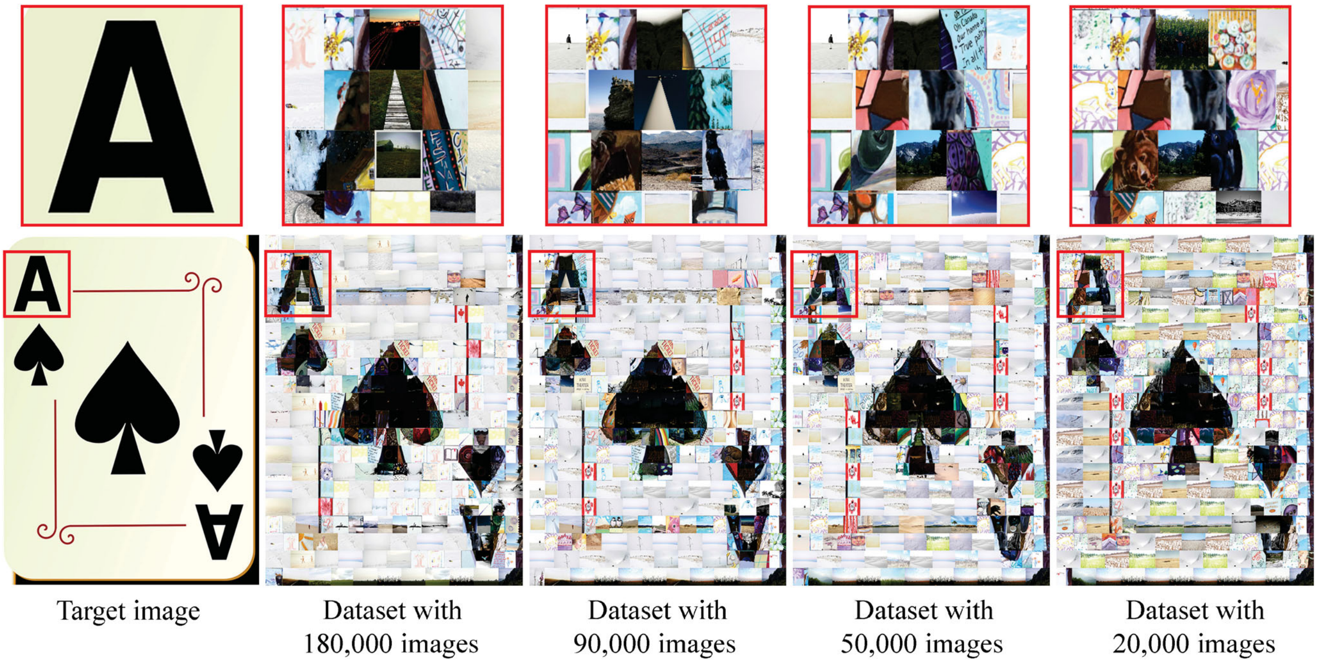
Target image
Dataset with 180,000 images
Dataset with 90,000 images
Dataset with 50,000 images
Dataset with 20,000 images

Fig. 8 Image mosaics produced by our method with different image datasets. As the size of the dataset decreases, the quality of texture matching and edge matching in the produced image mosaic degrades.

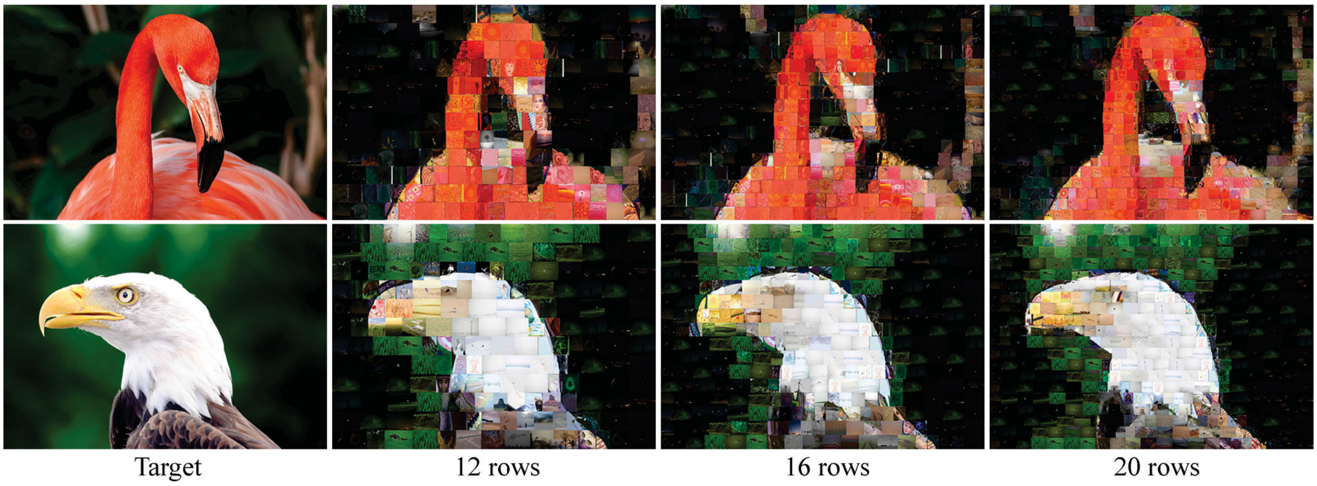

Fig. 9 Our method can produce high-quality image mosaics which preserve the edges of the target image, even when using few rows.

unacceptable, 3 adequate, and 5 perfectly acceptable. They were also asked about the recognizability of the replacement images during the study. After rating all image mosaic sets, the participants were required to state the factors they considered to produce their ratings.

Figure 10 shows a statistical summary of the user ratings. We can see that the image mosaics produced by our method have the highest scores in all sets. An ANOVA analysis also confirmed that there were significant differences ($p < 0.05$) between the scores using our approach and the other two approaches. Although image mosaic scores involve personal taste, these statistics still imply that our method generates more desirable results. It is also worth noting that some of our results have scores below 4, implying that the quality of the image mosaics produced could be further improved. As shown earlier, one simple solution is to include more images in the database.

The factors the participants considered important for rating often included smoothness and continuity of edges, especially of contours of objects in the image.

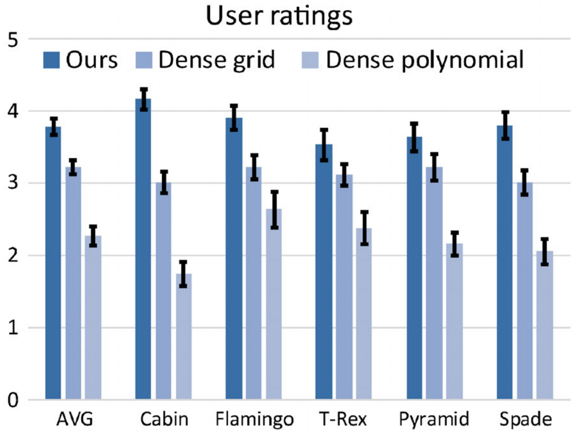

Fig. 10 User ratings for image mosaics produced using three different approaches. Error bars represent the standard error in the mean.

Some participants indicated that important features, such as the outlines of the cabin, should be faithfully recovered. Some participants preferred image mosaics with clean appearance, while other ones preferred diverse tiles. All claimed that they could recognize the content of each tile. This feedback indicated that our method is able to produce desirable image mosaic while keeping the recognizability of the tiles.

5 Conclusions, limitations, and future work

We have presented a novel method for producing *discernible* image mosaics with relatively large tiles.

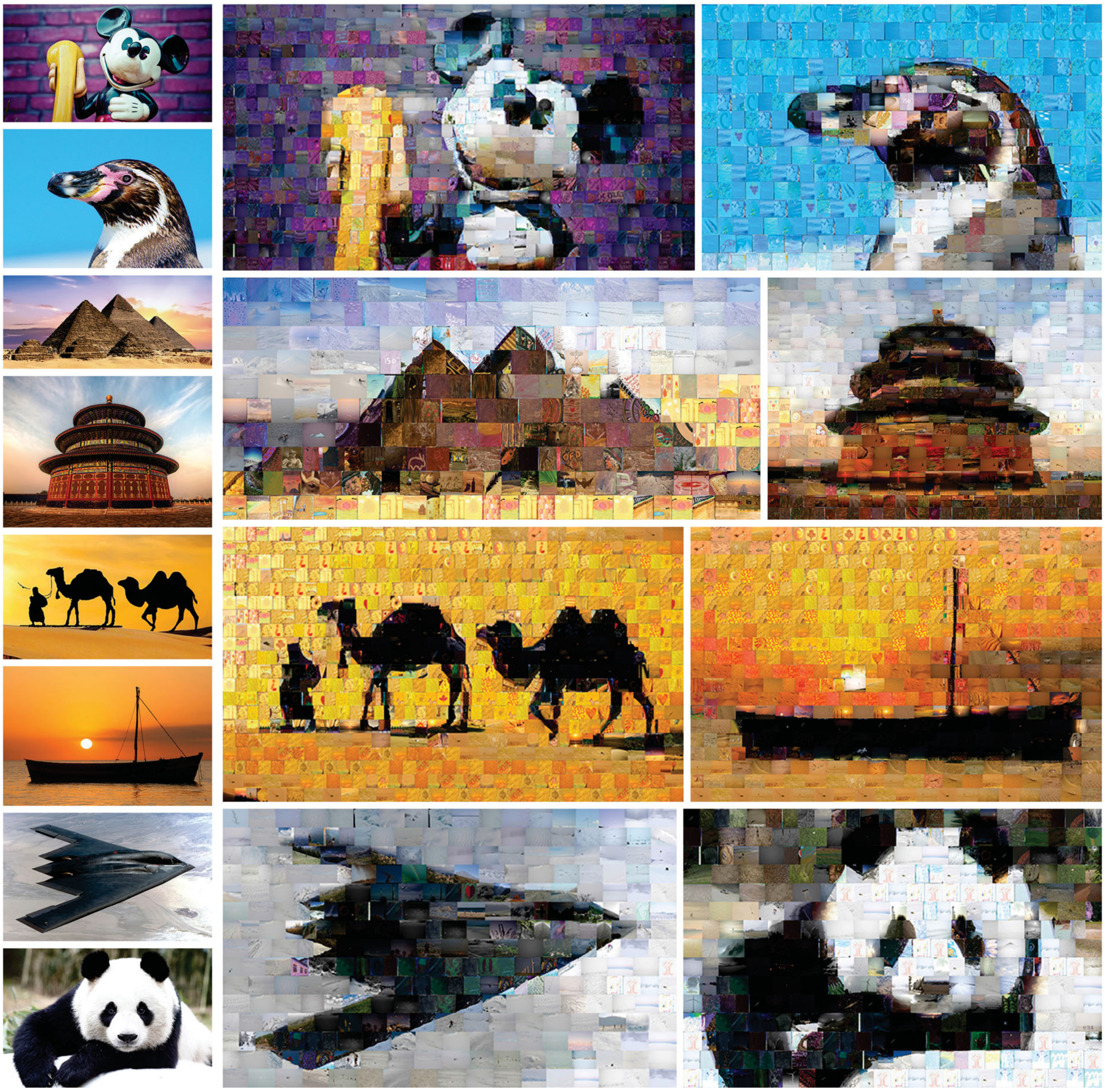

Fig. 11 Further image mosaics produced by our method.

Our method adopts an edge-aware image retrieval scheme, which emphasizes edge conformation between the query image and the retrieved images. The tile layout is adaptively determined via dynamic programming, based on the available images in the database and optimization of mosaic quality. Visual comparisons and user study results confirm that our method is able to produce image mosaics with better global resemblance to the target and local recognizability than previous works.

Nevertheless, with relatively large mosaic tiles, various forms of visual artifacts are still observable in most, if not all, results generated by our method. Insufficiently many photos in the database are always a contributing factor. Furthermore, our current implementation of edge-aware image retrieval is unable to handle soft or weak edges well. Salient features such as the eyes of the eagle in Fig. 9 play an important role in human perception but they are not handled via any special means in our method.

In addition to addressing the limitations mentioned above, we would also like to expand the adaptivity of the mosaic tiles. Possibilities include allowing both the heights and widths of the tiles to adapt, as well as photo transformations such as rotation and scaling. Incorporating image salience and semantics to improve the quality of mosaics are also natural paths to explore. For example, the eyes of the eagle could be recognized from the target photo so that we may offer the options of not replacing them or replacing them by more targeted photo retrieval.

Acknowledgements

We thank the anonymous reviewers and the editors for their valuable comments. This work was supported in part by the National Natural Science Foundation of China (Nos. 61602310, 61522213, and 61528208), Guangdong Science and Technology Program (No. 2015A030312015), Shenzhen Innovation Program (Nos. JCYJ20170302154106666, KQJSCX-20170727101233642), and NSERC (No. 611370).

References

[1] Wikipedia-contributors. Photographic mosaic. Available at `https://en.wikipedia.org/wiki/Photographic_mosaic`.

[2] Silvers, R.; Hawley, M. *Photomosaics*. Henry Holt and Co., Inc., 1997.

[3] Kim, J.; Pellacini, F. Jigsaw image mosaics. *ACM Transactions on Graphics* Vol. 21, No. 3, 657–664, 2002.

[4] Di Blasi, G.; Gallo, G.; Petralia, M. P. Smart ideas for photomosaic rendering. In: Proceedings of the Eurographics Italian Chapter Conference, Vol. 2006, 267–272, 2006.

[5] Orchard, J.; Kaplan, C. S. Cut-out image mosaics. In: Proceedings of the 6th International Symposium on Non-Photorealistic Animation and Rendering, 79–87, 2008.

[6] Pavić, D.; Ceumern, U.; Kobbelt, L. GIzMOs: Genuine image mosaics with adaptive tiling. *Computer Graphics Forum* Vol. 28, No. 8, 2244–2254, 2009.

[7] Miller, J.; Mould, D. Accurate and discernible photocollages. In: Proceedings of the 8th Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, 115–124, 2012.

[8] Zhang, L.; Ma, K.-L.; Yu, J. Adaptively tiled image mosaics utilizing measures of color and region entropy. In: Proceedings of the 9th International Symposium on Visual Information Communication and Interaction, 122–129, 2016.

[9] Battiato, S.; Di Blasi, G.; Farinella, G. M.; Gallo, G. Digital mosaic frameworks—An overview. *Computer Graphics Forum* Vol. 26, No. 4, 794–812, 2007.

[10] Dobashi, Y.; Haga, T.; Johan, H.; Nishita, T. A method for creating mosaic images using Voronoi diagrams. In: Proceedings of the Eurographics, Vol. 2, 341–348, 2002.

[11] Faustino, G. M.; de Figueiredo, L. H. Simple adaptive mosaic effects. In: Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing, 315–322, 2005.

[12] Zhang, L.; Yu, J. Image mosaics with irregular tiling. In: Proceedings of the 12th International Conference on Computer-Aided Design and Computer Graphics, 155–162, 2011.

[13] Lai, Y.-K.; Hu, S.-M.; Martin, R.-R. Surface mosaics. *The Visual Computer* Vol. 22, Nos. 9–11, 604–611, 2006.

[14] Hausner, A. Simulating decorative mosaics. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 573–580, 2001.

[15] Liu, Y.; Veksler, O.; Juan, O. Generating classic mosaics with graph cuts. *Computer Graphics Forum* Vol. 29, No. 8, 2387–2399, 2010.

[16] Battiato, S.; Di Blasi, G.; Gallo, G.; Guarnera, G. C.; Puglisi, G. Artificial mosaics by gradient vector flow. In: Proceedings of the Eurographics (Short Papers), 53–56, 2008.

[17] Wu, H.; Lyu, X.; Wen, Z. Automatic texture exemplar extraction based on global and local textureness measures. *Computational Visual Media* Vol. 4, No. 2, 173–184, 2018.

[18] Barnes, C.; Zhang, F. L. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media* Vol. 3, No. 1, 3–20, 2017.

[19] Barnes, C.; Zhang, F.-L.; Lou, L.; Wu, X.; Hu, S.-M. Patchtable: Efficient patch queries for large datasets and applications. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 97, 2015.

[20] Di Blasi, G.; Gallo, G.; Petralia, M. Puzzle image mosaic. In: Proceedings of the IASTED/VIIP, 33–37, 2005.

[21] Kwan, K. C.; Sinn, L. T.; Han, C.; Wong, T. T.; Fu, C. W. Pyramid of arclength descriptor for generating collage of shapes. *ACM Transactions on Graphics* Vol. 35, No. 6, Article No. 229, 2016.

[22] Huang, H.; Zhang, L.; Zhang, H.-C. Arcimboldo-like collage using internet images. *ACM Transactions on Graphics* Vol. 30, No. 6, Article No. 155, 2011.

[23] Reinert, B.; Ritschel, T.; Seidel, H.-P. Interactive by-example design of artistic packing layouts. *ACM Transactions on Graphics* Vol. 32, No. 6, Article No. 218, 2013.

[24] Zou, C.; Cao, J.; Ranaweera, W.; Alhashim, I.; Tan, P.; Sheffer, A.; Zhang, H. Legible compact calligrams. *ACM Transactions on Graphics* Vol. 35, No. 4, Article No. 122, 2016.

[25] Hu, S.-M.; Zhang, F.-L.; Wang, M.; Martin, R. R.; Wang, J. PatchNet: A patch-based image representation for interactive library-driven image editing. *ACM Transactions on Graphics* Vol. 32, No. 6, Article No. 196, 2013.

[26] Zhang, F.-L.; Wang, J.; Shechtman, E.; Zhou, Z.-Y.; Shi, J.-X.; Hu, S.-M. PlenoPatch: Patch-based plenoptic image manipulation. *IEEE Transactions on Visualization and Computer Graphics* Vol. 23, No. 5, 1561–1573, 2017.

[27] Rother, C.; Bordeaux, L.; Hamadi, Y.; Blake, A. AutoCollage. *ACM Transactions on Graphics* Vol. 25, No. 3, 847–852, 2006.

[28] Yu, Z.; Lu, L.; Guo, Y.; Fan, R.; Liu, M.; Wang, W. Content-aware photo collage using circle packing. *IEEE Transactions on Visualization and Computer Graphics* Vol. 20, No. 2, 182–195, 2014.

[29] Goferman, S.; Tal, A.; Zelnik-Manor, L. Puzzle-like collage. *Computer Graphics Forum* Vol. 29, No. 2, 459–468, 2010.

[30] Oliva, A.; Torralba, A.; Schyns, P. G. Hybrid images. *ACM Transactions on Graphics* Vol. 25, No. 3, 527–532, 2006.

[31] Chu, H.-K.; Hsu, W.-H.; Mitra, N. J.; Cohen-Or, D.; Wong, T.-T.; Lee, T.-Y. Camouflage images. *ACM Transactions on Graphics* Vol. 29, No. 4, Article No. 51, 2010.

[32] Tong, Q.; Zhang, S.-H.; Hu, S.-M.; Martin, R. R. Hidden images. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, 27–34, 2011.

[33] Farbman, Z.; Fattal, R.; Lischinski, D.; Szeliski, R. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics* Vol. 27, No. 3, Article No. 67, 2008.

[34] Muja, M.; Lowe, D. G. Flann: Fast library for approximate nearest neighbors. 2009. Available at `http://www.cs.ubc.ca/research/ann/`.

[35] Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 886–893, 2005.

[36] Zitnick, C. L.; Dollár, P. Edge boxes: Locating object proposals from edges. In:*Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693.* Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 391–405, 2014.

[37] Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 40, No. 6, 1452–1464, 2018.

[38] Rapid-Mosaic. Foto-mosaic-edda. Available at `https://fmedda.com/en/home`.

Pengfei Xu is an assistant professor in the Visual Computing Research Center at the College of Computer Science and Software Engineering, Shenzhen University. He received his bachelor degree in math from Zhejiang University in 2009, and his Ph.D. degree in computer science from Hong Kong University of Science and Technology in 2015. His primary research lies in human computer interaction and computer graphics.

Jianqiang Ding is a master student with the Visual Computing Research Center at the College of Computer Science and Software Engineering, Shenzhen University. He received his bachelor degree from Harbin Engineering University in 2015. His research interest is in computer graphics.

Hao Zhang is a full professor in the School of Computing Science at Simon Fraser University, where he directs the GrUVi lab. He obtained his Ph.D. degree from the University of Toronto, and both M.Math. and B.Math. degrees from the University of Waterloo. His research is in computer graphics with a focus on geometry modeling, shape analysis, 3D content creation, and computational design and fabrication.

Hui Huang is a distinguished professor at Shenzhen University, where she directs the Visual Computing Research Center. She received her Ph.D. degree in applied math from the University of British Columbia in 2008 and another Ph.D. degree in computational math from Wuhan University in 2006. Her research interests are in computer graphics and scientific computing, focusing on point-based modeling, image processing, geometric analysis, 3D acquisition and creation.