Adaptively Tiled Image Mosaics Utilizing Measures of Color and Region Entropy

Lina Zhang
The State Key Lab of CAD&CG
Zhejiang University
Hangzhou, 310027, China

Kwan-Liu Ma
Department of Computer Science
University of California-Davis
2063 Kemper Hall

Jinhui Yu
The State Key Lab of CAD&CG
Zhejiang University
Hangzhou, 310027, China

ABSTRACT

Image mosaicing involves splitting an input image into a set of tiles, then replacing each tile with another image from a large dataset so that, when viewed from a distance, the resulting image resembles the original. We present a new approach for generating image mosaics using variable sized tiles made up from patches taken from photographs, paintings and texture images. This is different from previous work, where either simple regular tiling or adaptive tiling based on variations of RGB color was used. We propose an adaptive tiling theme by means of region entropy. In order to avoid the mismatch in roughness between the sub-image in the tile region of the input image and tile images in the dataset that may arise in the previous RGB color based image descriptors, we introduce the region entropy into the image descriptor to achieve better matching in both color and roughness. We also propose a new metric to measure the quality of the image mosaic which takes both the similarity and the mutual information between the generated mosaics and input images into account. The final mosaic images in this work are obtained by optimizing an objective function based on this metric.

CCS Concepts

•Computing methodologies → Image-based rendering;

Keywords

Mosaic; tiling; entropy; information evaluation

1. INTRODUCTION

The mosaic is defined as an ancient art form usually made by arranging small pieces of stone or glass to create a picture or pattern. Mosaics may use either regular tiles such as cubic stones, or irregular shapes and sizes of ceramic, porcelain, glass, and stone for greater design variety. In addition to the tile shapes, tile colors are also chosen to depict features in the picture or structures in the pattern. In computer generated image mosaics, various elements such as photographic images, icons, and even a class of objects such as fruits and flags can be used as tiles. Compared with traditional mosaic tiles, image mosaic tiles may convey some additional meaningful visual information to the viewers, thus they enrich the expressive power of mosaics and have many applications for artistic and commercial purposes.

In this paper we propose a method to generate image mosaics with variable sized tiles composed out of photographs, patches of paintings and texture images. The right two figures in Fig. 1 are two examples of image mosaics of Beethoven's portrait [25] generated by our system. Here we address the problem of creating image mosaics as follows: Given a collection of tile images T and a target image I, construct an image mosaic M with variable sized tiles containing images selected from T which will resemble I when viewed from a distance. The main contributions of our method which distinguish our work from previous approaches are as follows:

(1) Most image mosaics use equally spaced rectangular image tiles (Fig. 2 (a) and (b)). A few use rectangular image tiles which change their sizes adaptively in the salient and non-salient regions as detected by the variations of RGB colors in I (Fig. 2 (c)). Instead, we achieve an adaptive tiling by merging neighboring tiles in the square grid based on region entropy (Fig. 2 (d)).

(2) Currently, color information is used to match a tile region in I and tile images in T in most image mosaic systems. However, such color based image descriptors may lead to a mismatch in roughness between the two (Fig. 5 *left*), so we introduce an image descriptor that takes both color and region entropy into account and use it to match the tile region in I and tile images in T (Fig. 5 right three images).

(3) In contrast to current approaches involving the evaluation of the fitness between M and I using distance metrics which focus on a single perspective only, we propose a metric in terms of both similarity and mutual information between M and I to evaluate the image mosaic and use it to optimize an objective function to obtain the final mosaic image.

2. RELATED WORK

Image mosaicing methods aim at generating artistic mosaic images. Battiato et al. presented an overview of mosaic techniques [3]. The earliest example of making mosaic effects in which each tile has a single bit of color came from an artistic filter in Photoshop [1]. Silvers used a large dataset

VINCI '16, September 24-26, 2016, Dallas, TX, USA

DOI: http://dx.doi.org/10.1145/2968220.2968228

Figure 1: From left to right: Oil portrait of Beethoven, mosaic with tiles images composed of music instruments and performers and mosaic with tile images of paint brush textures. Readers should note that the mosaics in this paper are usually best viewed on a color display at 300% zoom at a distance of 1.5 to 2.5 meters.

of images to create photo-mosaics with very impressive results [23]. Later on, Finkelstein and Range applied simple color shifting and scaling on the final image mosaic to better suggest the overall form [9]. Elber et al. tried to simulate traditional mosaics with cubic tiles of constant colors [8, 4].

In addition to the regular arrangement of tile images, Di Blasi et al. first introduced adaptive mosaics [6]. A different adaptive tiling method called *gizmos* was presented in [20]. Several researchers have developed methods to generate mosaic images with irregular tiles. Hausner arranged tiles of constant color using a centroidal Voronoi diagram and a distance field derived from user-specified contours to simulate decorative mosaics [10]. Kim and Pellacini introduced a general framework for creating so-called Jigsaw image mosaics (JIM) [11]. Battiato et al. rendered different mosaic styles automatically depending on artistic techniques considered such as *opus musivum* or *opus vermiculatum*, etc. [2]. Orchard and Kaplan introduced cut-out image mosaics [16]. Liu et al. formulated the mosaic simulating problem in a global energy optimization framework [15].

In addition to still mosaic image generation, several researchers focused on animated mosaics [10, 11, 24, 28, 13, 12]. Some others were interested in 3D mosaicing [14, 17, 18, 19, 5].

3. SYSTEM OVERVIEW

In image mosaics the most frequently used tiles are squares. With equal spacing of regular tiles (Fig. 2 (a) and (b)) one has to choose much smaller tiles in order to depict features in I well in M, thus making images in tiles difficult to see unless they are enlarged many times. Equal spacing of regular tiles also tends to have visual periodicity. Adaptive tiling (Fig. 2(c) and (d)) can avoid such periodicity by placing smaller tiles in the salient regions and larger tiles in non-salient regions.

In this work we propose a method to achieve adaptive tiling through bottom-up merging. The basic architecture of our system is presented in Fig. 3. The input I is split into a grid with small tiles which are then merged as guided by region entropy to obtain the resultant tiling pattern. Once the tiling pattern is obtained, we match the tile regions in I and tile images in T using a 28-dimensional image descriptor to generate a temporal mosaic image M_t, from which we subsequently calculate the similarity and mutual information between M_t and I, and use them to optimize an objective function to determine the final image mosaic M (cf. Section 5.1) .

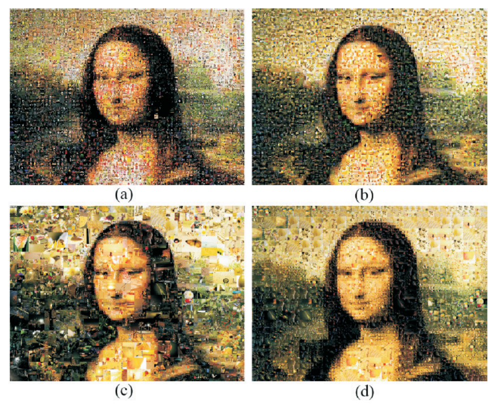

Figure 2: Comparison of our method with (a) a solution created with the *Patchworkr* tool [31], (b) a regular solution and (c) an adaptive solution with *gizmos* [20], and (d) a solution using our adaptive method. T=2700 in (a) and (b), T=1180 in (c) and T=1430 in (d).

4. ADAPTIVE TILING BY BOTTOM-UP MERGING

Compared with equally-spacing tiling, adaptive tiling has the following characteristics: the image mosaic can be made using fewer tiles, smaller tiles are able to preserve features in M and bigger tiles in the non-salient regions allow a better perception of the images in the tiles.

Since salient regions in I contain structures and patterns which transmit more information than those in non-salient regions, we use Shannon's information entropy [22] as a metric for evaluating the degree of information transmitted in the tiles. The concept of information entropy describes how

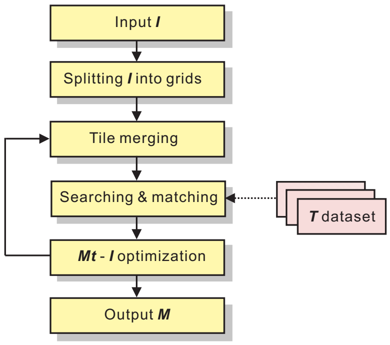

Figure 3: System workflow overview.

much uncertainty (information) there is in a signal or image. In an 8-bit gray-scale image, the intensity entropy can be calculated as:

$$H = -\sum_{i=0}^{255} p_i \log_2 p_i \tag{1}$$

where p_i is the probability that a random pixel chosen from the image has an intensity i. This can be approximated by use of image's histogram in the vicinity of the image. That is, for each grey level or bin in the histogram, we compute the frequency

$$p_i = \frac{number\ of\ pixels\ in\ bin(i)}{total\ pixels\ in\ image} \tag{2}$$

When I is a color image, we convert it into a gray-scale image and calculate its intensity entropy. In our system we use Eq. 1 to calculate the region entropy of sub-images in the tiles in I. A larger entropy value corresponds to an image texture or edge region, while a smaller entropy value corresponds to smooth image area. We further normalize all region entropy values so that a parameter $h \in [0, 1]$ can be used as a threshold in the following bottom-up merging, and h is determined by optimizing an objective function, as detailed in Section 6.

In our bottom-up merging, the target image I is first split into a square grid, the size of each square is initially set to $S_r \times S_r$ pixels (in the rest of the paper we omit the unit 'pixel' for image and tile sizes). From experiments we found that, for images of size $(m \times n)$ where $768 \leq m, n \leq 1024$ (which are typical of image sizes used in the examples given in this paper), $S_r = 15$ is adequate for the preservation of target object features in M. Then, each group of four adjacent tiles is merged to obtain a bigger tile with size $2S_r \times 2S_r$ if their region intensity entropy is smaller than h. The process is repeated until no more tiles can be merged. Next, we repeat the process to each group of four adjacent bigger tiles. We note that unmerged small tiles in the previous step may be left alone. Usually two repetitions of merging is enough for adaptive tiling with a moderate range of tile sizes, because very large tiles would significantly ruin the visual effects of M. In Fig. 4 we show the merging process with tile pattern variations during the objective optimization.

5. TILE IMAGE MATCHING AND COLOR CORRECTION

In image mosaics there are many possible choices for T, photographic images, paintings, icons, textures, even abstract patterns. The number of images in T involved in existing image mosaic systems may range from dozens to a few thousand, and even to a million [20]. Both larger and smaller T datasets have advantages and disadvantages from different perspectives.

First, a small T may be made up so that the content in T may have some associations with the target image I for artistic and commercial purposes. For instance, we can select some segments from Picasso's paintings [21] as T to generate a mosaic showing a portrait of Picasso himself (Fig. 7). When using a very large T it is almost impossible to make the content in T associate to I in a particularly meaningful manner.

Secondly, a large T has a sufficient range of luminances to cover the range of luminances in I, so we do not need to make color correction on tile images in mosaic generation. As a smaller T contains a limited range of luminances and so may not cover the range of luminances in I here, color correction is then needed for a better visual matching.

Thirdly, a large T requires more time to search and match than a smaller T does. For instance, a few minutes are needed to generate mosaics in *gizmos* [20] with tile images near to a million. Whatever kind of T is used, matching between tiles in I and images in T must be performed during mosaic generation.

5.1 Matching

When selecting image tiles aesthetic criteria predominate over technical ones and implementers are free to experiment. Our particular choice of tile matching mechanism is primarily based on the very practical method given in [6], in which each tile image is partitioned into a 3×3 grid and for each grid cell the average RGB color is computed. This leads to a 27-dimensional image descriptor. However this RGB color based image descriptor does not take the region roughness into account. As a result, a tile image with a high roughness value but similar average colors to the tile region in I may be selected from T to paint a tile region with a high smoothness value, as indicated by Fig. 5 *left*, where Beethoven's smooth face is covered by a few tile images with distinct variations in roughness. Such mismatches in roughness between the tiles in I and images in T is visually not desirable in image mosaics.

Since entropy is the metric which is most helpful in determining the roughness in an image, we introduce an additional dimension, the tile entropy H, to measure the region roughness in the image descriptor in order to obtain a better matching of both color and roughness. The match between the tile in I and image in T is then achieved by minimizing the quadratic functional of the following 28-dimensional image descriptor (ImD):

$$\begin{aligned} ImD(H, C) = {} & \alpha\left(\frac{H_S - H_T}{H_{max}}\right)^2 + \\ & \beta \left\{ \frac{1}{3 \times \mathbf{K}(V_{I \cap T})} \sum_{C \in \{R,G,B\}} \sum_{i \in V_{I \cap T}} \left(\frac{C_{S,i} - C_{T,i}}{C_{max}}\right)^2 \right\} \end{aligned} \tag{3}$$

where $V_{I \cap T}$ contains the index pairs of the matched vertices in $I \cap T$, $\mathbf{K}(V_{I \cap T})$ is the cardinality of $V_{I \cap T}$ (=9 here), $H_{max} = 8$, $C_{max} = 255$ and the S are the matching vertices

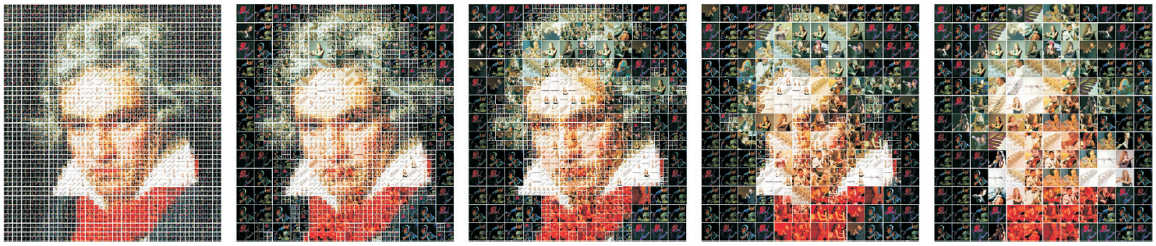

Figure 4: Variations of the tiling pattern as entropy threshold h increases.

Figure 5: Left: A mosaic generated using an RGB image descriptor. The remaining three mosaics are generated with our image descriptor of parameter settings on α/β=0.8/0.2, 0.6/0.4 and 0.4/0.6, respectively.

for I in $I \cap T$. Also α and β are factors which weight the region entropy relative to the 27-dimensional RGB color.

To show the effect of entropy in our image descriptor, we include some smooth brush stroke texture images in T for Fig. 5 *left* and vary the settings for α and β in Eq. 3 to generate image mosaics, Fig. 5 shows the effect on the image mosaics of increasing α while decreasing β from left to right. We can see that decreasing α results in inhibition of roughness matching in favor of color matching, while increasing β leads to the opposite. Consequently, progressively more smooth textures are selected by our image descriptor to cover the smooth regions in I, as expected and seen in the right three images in Fig. 5. Other parameter settings and effects are possible.

In our system all tile images are square in shape. When adaptive tiling is used, the tile regions are also square in shape (although they vary in size). So we can use Eq. 3 directly to match an image from T, resize it to fit the tile region in I and copy the resized image into the tile region in M.

Using Eq. 3 our system can select the most suitable tile images to cover the inhomogenous regions of I. These images usually vary in content and their overall look is visually pleasing. Our system may also pick up the same tile image to cover some bigger homogeneous regions of I in which variations in color and roughness are naturally very low, but this is undesirable in image mosaics from the artistic point of view. We therefore pick up tile images with the three smallest values of $ImD(H, C)$ from T and then randomly select between them when tiling such regions in I.

5.2 Color correction

In our system we adopted a small dataset T with tile image numbers ranging from 50 to 100, which allows users to construct their own dataset to meet their artistic demands more easily. Although in the matching phase our system can select tile images with a balance of both color and roughness from T, adoption of a small tile dataset in general could not ensure a good match between tile images and T in color, thus color correction on tile images is required after matching. We choose the algorithm given in [9] to correct colors in tile images because it is sufficient for our purposes and easy to compute.

6. OBJECTIVE FUNCTION

In this work image mosaics are generated with an objective function, which is derived from evaluation on the fitness between M and I by taking into account both the root mean square difference and the mutual information between M and I. Although many approaches have been proposed for generating image mosaics, only a few have addressed the evaluation of mosaic quality with metrics and these normally focus on a single perspective. D'Souze et al. evaluated their fitness by summing the differences in pixel grey levels between M and I in [7].

As the tile images in M do not just depict the target image I, but also convey additional information of themselves to the viewers on their own, we think that, in addition to the similarity in terms of RGB color distance between M and I, it is preferable to take the mutual information between M and I into account for the evaluation of these image mosaics. The mutual information $I_m(M; I)$ between M and I is defined as follows:

$$I_m(M; I) = H(M) + H(I) - H_j(M; I) \tag{4}$$

where $H(M)$ and $H(I)$ are the entropy of M and I, respectively, and $H_j(M; I)$ is the joint entropy of M and I, which is calculated by

$$H_j(M, I) = -\sum_{i=0}^{255}\sum_{j=0}^{255} p_{i,j} \log_2 p_{i,j} \tag{5}$$

where $p_{i,j}$ is the joint probability of finding the same intensity in both M and I, which we can approximate by using the joint histogram of M and I, a two dimensional array in which each entry is the likehood that an intensity i in M corresponds to an intensity j in I.

As mentioned already in Section 4, the choice of entropy parameter h for merging tiles will affect the results of adaptive tiling, which in turn also affects the final visual effects in M in terms of the information in T being transmitted to M. Intuitively, if fewer tiles are merged when h is small, more tiles with smaller sizes are left in M so that M would look visually more similar to I and, correspondingly, the mutual information shared by M and I would be bigger. Alternatively if more tiles are merged when h is larger, the mutual information reduces because more additional information from T is transmitted with bigger tiles, correspondingly the resultant M would become more dissimilar to I.

In order to confirm our intuition, we generated a series of mosaic images for Beethoven's portrait with h varying from small to large values. Fig. 4 shows five images taken from the generated mosaics with tile images related to musical instruments and performers. We constructed a number of datasets T for the different examples as described in Section 7, as presented here, and corresponding tiling patterns are drawn on them. Next, we calculated the root mean square difference $D(M;I)$ between M and I for varying h values in terms of RGB colors as follows:

$$D(M;I) = \| M - I \| = \sqrt{\sum_{C \in R,G,B} (C_M - C_I)^2} \quad (6)$$

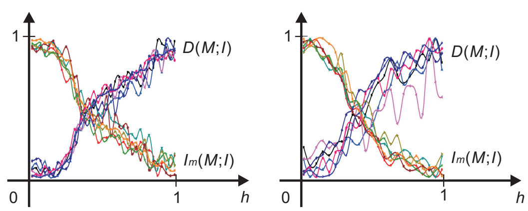

Figure 6: Left: Curves of $D(M;I)$ and $I_m(M;I)$ calculated using Beethoven's portrait and a different set of tile images. Right: Curves of $D(M;I)$ and $I_m(M;I)$ calculated using Monroe's portrait and a different set of tile images.

We likewise calculated the values of $I_m(M;I)$ for corresponding values of h for different sets of T. Fig. 6 *left* plots two sets of normalized $D(M;I)$ and $I_m(M;I)$ curves against h. We note that, although two curves show high frequency decorrelations, the general trends of $D(M;I)$ and $I_m(M;I)$ increases or decreases monotonically with h, which agrees with our intuition. When the target image was replaced with Marilyn Monroe's portrait [27], we got similar results of $D(M;I)$ and $I_m(M;I)$ calculated with different set of tile images, as shown in Fig. 6 *right*.

We can now see from Fig. 6 that both $D(M;I)$ and $I_m(M;I)$ are functions of h. To balance the similarity and the mutual information between M and I in the final image mosaics, we seek to optimize the objective function by minimizing the distance between $D(M;I)$ and $I_m(M;I)$:

$$H_w = \arg\min_h \|D(M(h);I(h)) - I_m(M(h);I(h))\| \quad (7)$$

where $arg(\cdot)$ returns to H_w the value of h which gives the minimum distance between $D(M;I)$ and $I_m(M;I)$.

The use of the minimization function in Eq. 7 requires the normalization of values for these functions. Since $D(M;I)$ is calculated using the differences in the RGB colors between M and I, it can be normalized easily by dividing through the maximum values of RGB. As for normalizing $I_m(M;I)$, we calculate the maximum value I_{max} of $I_m(M;I)$ for the case of no tiles being merged and the minimum value I_{min} for the case of all 4×4 tiles with size S_r being merged, and then normalize all $I_m(M;I)$ values obtained for the minimization iterations using $(I_m(M;I) - I_{min})/(I_{max} - I_{min})$.

In our implementation, the iteration step is taken equal to 0.005 to ensure the minimum difference between $D(M;I)$ and $I_m(M;I)$ is covered, and thus 200 iterations in total are needed for the variable h to cover the range [0,1]. Our experiments show that, for different input target images and T datasets, the minimum difference between $D(M;I)$ and $I_m(M;I)$ can be captured after 120 iterations. So, on average, it may take 115 seconds to complete the incremental optimization stage in our system, which is implemented with Visual C++ 2005 and run on a PC with Pentium 2.8GHz CPU and 2GB memory. The incremental iteration is shown dynamically in the accompanying video.

7. RESULTS

We present several results of image mosaics using this objective function. The datasets of tile images we constructed can be roughly divided into two categories, the relatively 'rough' ones composed of photographs or painting patches collected in some association with the input I, and 'smooth' ones composed of paint brush strokes and stone textures that can be used without specific association with I, as demonstrated in the examples given below.

In the first example we took Beethoven's oil painting portrait as I and collected some photos of musical instruments and performers from the internet to construct the dataset T, as shown in Fig. 1 *middle*. With this dataset we are conveying the idea that Beethoven's music is always with us, and will endure for generations. Fig. 1 *right* shows a mosaic image with tiles composed of patches of brush stroke textures taken from oil paintings.

In the second example we took Picasso's self portrait as I, and patches from Picasso's paintings and stone textures as the dataset T, as shown by the right two images in Fig. 7.

In Fig. 8 we show our third example in which the input is Marilyn Monroe's portrait in gray-scale, while the dataset T is constructed form a collection of her color photographs on the internet. In this example we convert color photographic tile images into gray-scale ones for the matching functions, while retaining the colors for tiling. Using color photographic tile images to depict a gray-scale input image brings a unique quality to the resultant image mosaics.

Fig. 9 shows the fourth example in which the target image is a photograph of the Japanese outdoor bronze statue, the Great Buddha of Kamakura [30]. We construct the dataset T by taking patches from Thang-ka [29], a unique style of religious scroll painting developed in the 7th century in the Tibetan area. By this example we can see that image mosaic is an exciting art form to integrate other art forms together, and many more combinations of different kind of art forms can be explored to produce image mosaics with a variety of new looks.

Figure 7: From left to right: the input image of Picasso's self portrait, mosaic with tiles of painting patches taken from Picasso's paintings, and mosaic with tiles of stone textures.

Figure 8: From left to right: the input photograph of Marilyn Monroe, mosaic with tiles of photographic patches of Monroe, and mosaic with tiles of paint brush stroke textures.

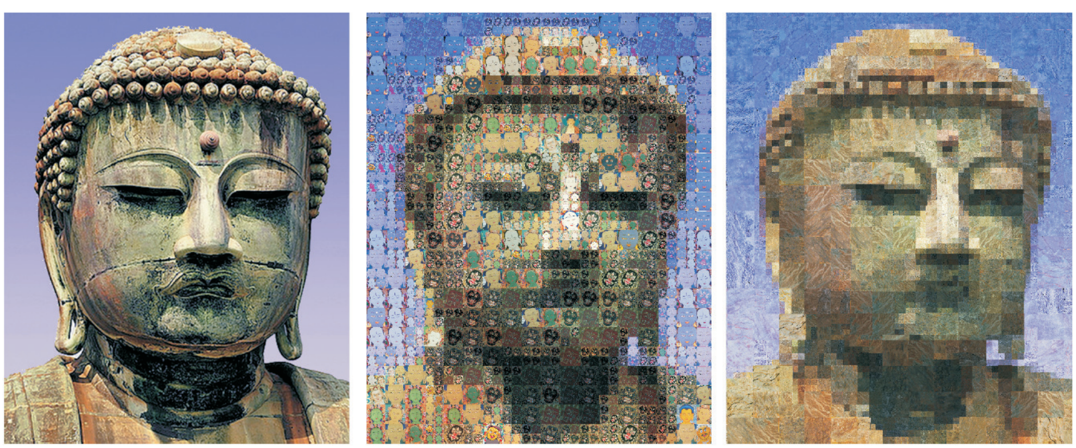

Figure 9: From left to right: the input photograph of Great Buddha, mosaic with tiles of patches taken from Thang-Ka paintings,and mosaic with tiles of stone textures.

Table 1 presents the image size, tile number (TN), normalized $D(M;I)$ and $I_m(M;I)$ as denoted by $TN/D/I_m$ in the table for the dataset *RoughT* taken from photos or paintings and *SmoothT* from painting stroke or stone textures.

On the bottom row of the table we show the averages of

Table 1: Summary of image size, tile number, $D(M;I)$ and $I_m(M;I)$ for previous examples.

	Image size	RoughT $TN/D/I_m$	SmoothT $TN/D/I_m$
Beethoven	912×960	1583/.257/.555	1502/.143/.876
Picasso	768×960	1922/.328/.287	1679/.137/.939
Monroe	768×960	1781/.296/.431	1886/.207/.584
Buddha	768×960	1139/.293/.356	1146/.143/.876
Average		1606/.294/.407	1553/.157/.819

the tile number, average $D(M;I)$ and $I(M;I)$ measures for different tile images. Where relatively smooth tile images such as painting brush stroke or stone textures are used, the average RGB distance between M and I is less than that for tile images composed of patches taken from photos and paintings. This is because smooth tile images tend to depict smooth areas on faces better than the relatively rough tile images composed of photos and paintings. Also, the average of the mutual information measure is maximized because smooth tile images contain less information than those taken from photos and paintings.

Finally we compare our method with other previous solutions by using two examples with RGB distance $D(M;I)$ and mutual information $I_m(M;I)$. The first example is Mona Lisa (Fig. 2) [26] and the second is the Chinese Yinyang Taji pattern (Fig. 11). We have not included the original input images of Mona Lisa and the Taiji pattern because they are well known.

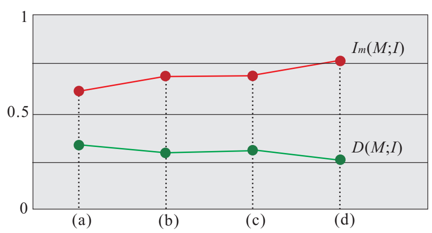

Figure 10: The mutual information and RGB distance measures between (a),(b),(c) and (d) in Fig. 2.

Fig. 10 plots $D(M;I)$ and $I_m(M;I)$ corresponding to mosaics of Mona Lisa generated with different methods. We note that both regular and adaptive tiling (Fig. 10 (b) and (c)) in *gizmos* have smaller RGB distances than that in *Patchworkr* [31] (Fig. 10 (a)), which indicate that the polynomial descriptors in *gizmos* achieve better matching than the *Patchworkr* tool does. The smaller RGB distance from our method (Fig. 10 (d)) indicates our color and region entropy based image descriptor is able to match more widely than the other methods we have compared. We note also, that the image mosaics generated with polynomial descriptors in *gizmos* have almost the same measures of mutual information, and these are slightly bigger than the corresponding measures for the *Patchworkr* tool, but smaller than for our method.

Fig. 12 plots $D(M;I)$ and $I_m(M;I)$ corresponding to mosaics of the Taiji pattern generated by different methods. We note that the RGB distances corresponding to mosaics generated with feature and non-feature descriptors (d), adaptive

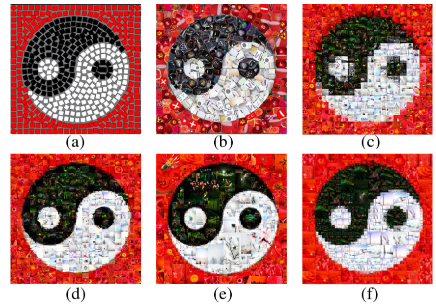

Figure 11: Comparison with (a) decorative mosaic [10] , (b) JIM [11],(c)-(e) with average color matching, feature and non-feature descriptors, and adaptive mosaicing in *gizmos* [20], and (f) using our method. For all results roughly 400 tiles were used. The images (a)-(b) were taken from [11], and (c)-(e) from [20].

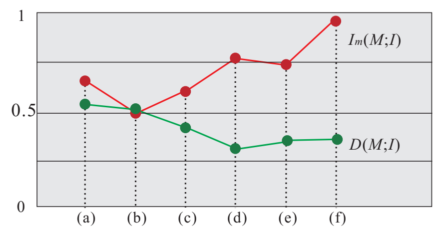

Figure 12: The mutual information and RGB distance measures between (a),(b),(c), (d), (e) and (f) in Fig. 11.

mosaicing in *gizmos* (e) and our color and region entropy descriptor (f) are smaller than that in the decorative mosaic (a), JIM (b) and average color match in *gizmos* (c). While the mutual information measure reaches its smallest value in JIM due to the different set of tile images used (mosaic tiles are not square in JIM). It is also interesting to note that, although the same tile images are used in (c),(d),(e) and (f), the image mosaics generated with our method have the biggest mutual information value among these four examples, and has nearly the same measure of RGB distance with adaptive mosaicing in *gizmos*.

8. CONCLUSIONS AND FUTURE WORK

Image mosaics are a digital age refinement on traditional mosaics and have many interesting applications. We have presented a system capable of generating image mosaics with adaptive tiling based on region entropy. A key issue associated with image mosaic generation using small datasets is the feature size matching between the tile images and input images. In our current work, a minimum tile size is suggested from experiments. A topic for future work is thus to find a method to measure the feature size for images of arbitrary objects, so that the minimum tile size can be determined automatically.

In our current image mosaic system, the tile image datasets are constructed manually, according to the associations that are intended to be built between the tile images and input images. It would be desirable for the tile image datasets to be constructed automatically or semi-automatically. This requires semantic recognition of classes of objects in the given images, which is another topic for future research.

9. ACKNOWLEDGMENTS

We thanks the support of the National Natural Science Foundation of China(No.61379069) and the Key Technologies R&D Program of China(No.2014BAK09B04).

10. REFERENCES

[1] Adobe. Photoshop. http://www.adobe.com.

[2] S. Battiato, G. D. Blasi, G. M. Farinella, and G. Gallo. A novel technique for opus vermiculatum mosaic rendering. In *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision(WSCG 2006)*, pages 133–140, 2006.

[3] S. Battiato, G. D. Blasi, G. M. Farinella, and G. Gallo. Digital mosaic frameworks - an overview. *Computer Graphics Forum*, 26(4):794–812, June 2007.

[4] S. Battiato, G. D. Blasi, G. Gallo, G. C. Guarnera, and G. Puglisi. Artificial mosaic by gradient vector flow. In *Proceedings of Eurographics*, 2008.

[5] S. Battiato and G. Puglisi. 3d ancient mosaics. In *Proceedings of ACM Multimedia*, pages 1571–1574, October 2010.

[6] G. D. Blasi, G. Gallo, and M. P. Petralia. Smart ideas for photomosaic rendering. In *Proceedings of Eurographics Italian Chapter Conference*, pages 267–272, 2006.

[7] D. D'Souza, V. Ciesielski, M. Berry, and K. Trist. Generation of self-referential animated photomosaics. In *Proceedings of the 15th international conference on Multimedia*, pages 489–492. TeX Users Group, March 2007.

[8] G. Elber and G. Wolberg. Rendering traditional mosaics. *The Visual Computer*, 19(1):67–78, January 2003.

[9] A. Finkelstein and M. Range. Image mosaics. In *Proceedings of Electronic Publishing, Artistic Imaging, and Digital Typography*, pages 11–12, 1998.

[10] A. Hausner. Simulating decorative mosaics. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 573–580, August 2001.

[11] J. Kim and F. Pellacini. Jigsaw image mosaics. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 657–664, 2002.

[12] K. Kim, I. Essa, and G. Abowd. Interactive mosaic generation for video navigation. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 655–658, 2006.

[13] A. Klein, T. Grant, A. Finkelstein, and M. Cohen. Video mosaics. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 21–28, June 2002.

[14] Y.-K. Lai, S.-M. Hu, and R. Martinn. Surface mosaics. *The Visual Computer*, 22(9-11):604–611, 2006.

[15] Y. Liu, O. Veksler, and O. Juan. Generating classic mosaics with graph cuts. *Computer Graphics Forum*, 29(8):2387–2399, 2010.

[16] J. Orchard and C. Kaplan. Cut-out image mosaics. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 79–87. ACM, 2008.

[17] V. A. D. Passos and M. Walter. 3d mosaics with variable-sized tiles. *The Visual Computer*, 24(7-9):617–623, July 2008.

[18] V. A. D. Passos and M. Walter. Simulation and rendering of opus palladium 3d mosaics. In *Proceedings of XXI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2008)*, pages 263–269, 2008.

[19] V. A. D. Passos and M. Walter. Opus palladium and mixed styles. *The Visual Computer*, 25(10):939–946, 2009.

[20] D. Pavic, U. Ceumern, and L. Kobbelt. Gizmos: Genuine image mosaics with adaptive tiling. *Computer Graphics Forum*, 28(8):2244–2254, December 2009.

[21] P. Picasso. Self-portrait. http://www.wikiart.org/en/pablo-picasso/self-portrait-1907, 1907.

[22] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379ĺC423, July 1948.

[23] R. Silvers and M. Hawley. *Photomosaics*. Henry Holt and Company, New York, 1997.

[24] K. Smith, Y. Liu, and A. Klein. Animosaics. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 201–208. ACM, July 2005.

[25] J. K. Stieler. A portrait of ludwig van beethoven. https://upload.wikimedia.org/wikipedia/commons/thumb/6/6f/Beethoven.jpg/800px-Beethoven.jpg, 1820.

[26] L. D. Vinci. Mona lisa. http://g.hiphotos.baidu.com/baike/pic/item/c8ea15ce36d3d5393086e9fd3b87e950352ab051.jpg, 1504.

[27] A. Warhol. Marilyn monroe. http://scm.ulster.ac.uk/~B00586593/des106/des_weeks/Art%20Family%20Tree/marilyn_monroe.jpeg, 1953.

[28] G. Wijesinghe, S. B. M. Sah, and V. Ciesielski. Grid vs. arbitrary placement of tiles for generating animated photomosaics. In *IEEE Congress on Evolutionary Computation*, pages 2734 – 2740. IEEE, June 2008.

[29] Wikipedia. Thangka. http://en.wikipedia.org/wiki/Thangka, 2010.

[30] J. Willoughby. Giant buddha of kamakura. https://farm8.staticflickr.com/7426/12443638163_4e5595cc55_b.jpg, 1953.

[31] A. Yavari. Patchworkr. http://www.fumpf.de/patchwork/.