

Research Article

Artificial Mosaic Generation with Gradient Vector Flow and Tile Cutting

Sebastiano Battiato, Alfredo Milone, and Giovanni Puglisi

Image Processing Laboratory, Dipartimento di Matematica e Informatica, University of Catania,
Viale A. Doria 6, 95125 Catania, Italy

Correspondence should be addressed to Giovanni Puglisi; puglisi@dmi.unict.it

Received 8 November 2012; Accepted 28 February 2013

Academic Editor: Filippo Stanco

Copyright © 2013 Sebastiano Battiato et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mosaics are images obtained by cementing together small colored fragments and are an ancient example of discrete primitive-based images. Artificial mosaics are illustrations composed by a set of small images called “tiles” that tessellate a source image aiming to reproduce the original visual information in a mosaic-like style. In this paper, we propose a mosaic generation technique based on gradient vector flow (GVF) together with a set of tile cutting heuristics evaluated according to aesthetic criteria. The effectiveness of the proposed approach has been confirmed by a series of tests and comparisons with state-of-the-art techniques.

1. Introduction

Art often provides valuable hints for technological innovations especially in the fields of image processing and computer graphics. Non-Photorealistic Rendering (NPR) is a successful area of computer graphics and it is nowadays applied to many relevant contexts: scientific visualization, information visualization, and artistic style emulation [1–4]. Within NPR, the recent approach to digitally reproduce artistic media (such as watercolors, crayons, and charcoal) and artistic styles (such as cubism, impressionism, and pointillism) has been gaining momentum and is very promising [3–7]. Several denominations have been proposed for this narrower area within NPR: artistic rendering (AR) [1] and computational aesthetics (CA) [8] are among the most popular ones. Its main goal can be resumed as follows: *to reproduce the aesthetic essence of arts by means of computational tools.*

The focus of this paper is related to novel methodologies for the automatic generation of ancient digital mosaics [9, 10]. Mosaics, in essence, are images obtained by cementing together small colored fragments. Likely, they are the most ancient examples of discrete primitive-based images [11, 12]. In the digital realm, mosaics are illustrations composed by a collection of small images called “tiles.” The tiles tessellate a source image with the purpose of reproducing the original

visual information rendered into a new mosaic-like style. The same source image may be translated into many strikingly different mosaics. Factors like tile dataset, constraints on positioning, deformations, and rotations of the tiles are indeed very influential upon the final results. As an example, the creation of a digital mosaic resembling the visual style of an ancient-looking manmade mosaic is a challenging problem because it has to take into account the polygonal shape and the size of the tiles, the need to pack the tiles as densely as possible, and, not least, the strong visual influence that tile orientation has on the overall perception of the mosaic. In particular, orientation cannot be arbitrary but it is constrained to follow the gestalt choices made by the author of the source picture. Tiles, hence, must follow and emphasize the main orientations chosen by the artist. In this work we extend our previous mosaic generation approach introducing some computational tools able to emulate the artistic “cut” often used by real mosaicists. This paper provides two kinds of cutting methods and parameters setting that control the cutting by making use of ad hoc analysis and experiments. Previous mosaic studies have focused on the decision of optimal position (high coverage of tiles, edge preservation, and minimization of overlap) of predefined shaped tiles. On the other hand, this study focuses on the tile cutting process, so that better tiling can be available. We have also performed

several experiments to numerically evaluate, with respect to the various involved parameters, some aesthetic criteria such as total coverage area and number of tiles.

The rest of this paper is organized as follows: Section 2 reviews the main digital mosaic generation techniques. Section 3 describes the proposed 2D approach, whereas Section 4 introduces the tile cutting heuristics. Finally, Section 5 reports the experimental results, and Section 6 is devoted to final discussions and suggestions for future works.

2. Related Works

The first attempt to reproduce a realistic ancient mosaic was presented by Hausner [13] who also proposed a mathematical formulation of the mosaic problem. Lots of artificial mosaic generation techniques have been then proposed (for a complete survey see [9, 10]). The key of any technique aimed at the production of digital ancient mosaics is the tile positioning and orientation. The methods proposed in the literature use different approaches to solve this problem, obtaining different visual results. Some techniques are based on a Centroidal Voronoi Diagrams (CVD) approach [13–15], whereas other methods [16–19] compute a vector field by making use of different strategies (i.e., graph cuts minimization [20], gradient vector flow [21]). Tile positioning is then performed with iterative strategies [13–16] or reproducing the ancient artisans style by using a “one-after-one” tile positioning [17, 22, 23]. A different nondeterministic approach is used in [24]. In the following we provide additional details about some techniques that are in some way related to the proposed one (cut strategy, vector field, etc.).

The algorithm presented in [22, 23], based on directional guidelines and distance transform, is able to lead to very realistic results by employing some image-processing techniques to obtain a precise tile placing. The authors, in order to obtain a high degree of similarity in terms of style with respect to ancient mosaics, also consider tile cutting.

A novel technique for ancient mosaics generation has been presented in [16] and further refined in [25]. Tile orientation field is generated considering the strong edges of the underlying image. Moreover, orientations are forced to vary smoothly in order to produce pleasing mosaics and reduce the gap between tiles. This field is obtained by using a global optimization approach (α -expansion algorithm) [20]. The packing of the tiles is then performed in two steps. First, a set of mosaic layers (M_1, M_2, \dots, M_n) is generated. Later, the mosaic layers are stitched together taking into account gap space minimization, the absence of broken tiles, and the crossing of strong edge intensity (no cutting is employed). This task is performed through the graph cuts algorithm [20].

Preliminary version of this paper has been published in [17]. Compared to [17], we have improved the algorithm including tile cutting capability by making use of proper heuristics together with a deep study about the impact of the involved parameters with respect to the overall quality of the generated mosaic. Moreover additional comparisons with state-of-the-art approaches, both in terms of quality and computational time, are now provided.

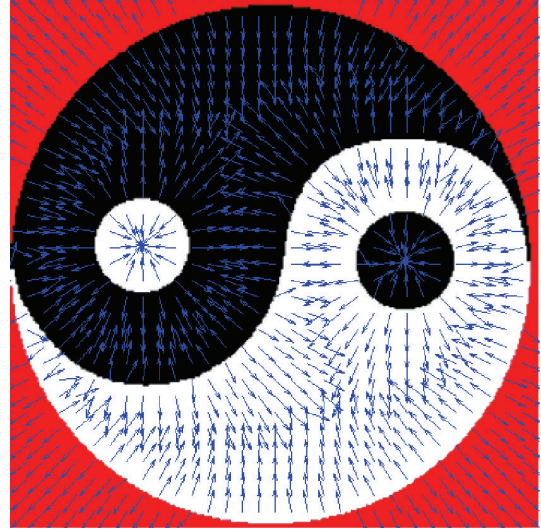


FIGURE 1: Input image and its corresponding GVF field.

3. 2D Mosaic Generation Approach

The proposed approach of artificial mosaic generation is based on the following two steps:

- (i) GVF (gradient vector flow) field computation-based on [21] algorithm;
- (ii) Rule-based tile positioning.

GVF is a dense force field designed by the authors of [21] in order to solve the classical problems that affect snakes: sensitivity to initialization and poor convergence to boundary concavity. Starting from the gradient of an image, this field is computed through diffusion equations. GVF is a field of vectors $\mathbf{v} = [v, u]$ that minimizes the following energy function:

$$E = \iint \mu (u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2, \quad (1)$$

where the subscripts represent partial derivatives along x and y axes, respectively, μ is a regularization parameter, and $|\nabla f|$ is the gradient computed from the intensity input image. Due to the formulation described above, GVF field values are close to $|\nabla f|$ values where this quantity is large (energy E , to be minimized, is dominated by $|\nabla f|^2 |\mathbf{v} - \nabla f|^2$) and are slow-varying in homogeneous regions (the energy E is dominated by the sum of the squares of the partial derivatives of GVF field). An example of GVF field is shown in Figure 1. This vector field can be then used to effectively drive tile positioning. Edge information is preserved, propagated in the close regions, and merged together in a smoothly way.

Let I be the input image to be mosaiced. In order to simplify the algorithm, we work only on the luminance channel of the image I , eliminating hue and saturation information. The luminance $L(I)$ is then equalized and the discrete gradient of the results is calculated, by means of crossing difference. The equalization process, especially for natural images, allows to normalize the overall gradient distribution. The horizontal and vertical components of the gradient $\nabla L(I)$

```

Input: A Raster Image  $I$ 
Output: Artificial Mosaic of Image  $I$ 
begin
   $L(I) = \text{Luminance}(I)$ 
   $G(I) = \text{Robert's Gradient}(\text{Equalize}(L(I)))$ 
   $[u, v] = \text{GVF}(|G(I)|_{\infty}, \mu, n\text{Iterations})$ 
   $gvf(I) = (u^2 + v^2)^{1/2}$ 
  Sort in queue  $Q$  pixels  $(i, j)$  according to decreasing
   $gvf(i, j)$  values. Only pixels whose  $gvf$  is greater
  than a threshold  $T$  go into  $Q$ .
  while  $Q$  is not empty do
    Extract pixel  $P$  from  $Q$ 
    Place a tile in  $P$  at angle
     $\alpha = \tan^{-1}(v(i, j)/u(i, j))$ 
    if in this way the tile overlaps with previously
    placed tiles then
      Skip tile positioning
  for  $j = 1$  to  $\text{length}(I)$  do
    for  $i = 1$  to  $\text{width}(I)$  do
      Place a tile in the pixel  $(i, j)$  at angle
       $\alpha = \tan^{-1}(v(i, j)/u(i, j))$ 
      if in this way the tile overlaps with
      previously placed tiles then
        Skip tile positioning
  end

```

ALGORITHM 1: Mosaic generation algorithm.

are used as input for the GVF algorithm. Notice that the gradient computation is performed using Robert's kernel [26]. This choice is more noise sensitive and hence incorporates in the final mosaic a little, aesthetically pleasant, randomness. All the tiles have the same rectangular shape, the same size, and do not overlap. Placing order is fundamental in terms of visual overall effect. Heuristically, we start positioning tiles at pixels with a high value of $|GVF(I)|$ and impose that the tile orientation is equal to the GVF direction in its central point. As a consequence, the areas close to the more relevant edges of the input images are taken care of first.

The second step of the algorithm is devoted to cover the homogeneous regions of the image. This is accomplished by simply placing each tile one by one following the order left to right, up to bottom, starting from the upper-left corner of the image. This heuristic strategy, somewhat arbitrary, is justified by the properties of the GVF: such technique leads to aesthetically pleasant results, by preserving main orientations and covering a wide portion of pixels with tiles densely packed. The algorithm is summarized in Algorithm 1.

4. 2D Mosaic Tile Cutting

Ancient mosaicists could make use of irregular tiles in the mosaic creation. Irregular tiles are suited to follow principal image edges, properly cover the image canvas obtaining hence visually pleasant mosaics. In our case, taking into account well defined ancient styles [12], both the tile size and its shape (i.e., rectangular) are designed at the beginning of the process. The final mosaic (with irregular tiles) is then

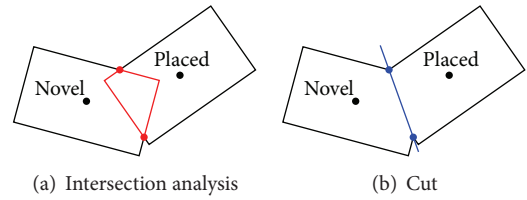


FIGURE 2: Shared cut.

obtained employing tile-merging and tile-cutting strategies as better specified below. Notice that a simple approach (to overcome the rectangular limitation) that considers little more jittered shapes with a random number of sides and side length, placing these random tiles in the image, cannot work in practice due to the minor percentage of covered area.

The original approach has been extended by considering two different strategies of tile cutting: subtractive and shared cut. The former cuts only the novel tiles; that is, tiles that are not already present in the mosaic; the latter cuts both novel and already placed tiles (shared cut actually unions two tiles and can be considered a tile-merging strategy). Both strategies together with the involved parameters are described in the following subsections. In particular when the tile to be placed overlaps with some already placed tiles (Algorithm 1), a series of cutting heuristics is properly introduced.

4.1. Shared Cut. Let tile_P and tile_N be the tile, already placed and to be placed (novel), respectively. Their intersection creates some novel vertexes placed on their border. The cutting is performed considering the line connecting these vertexes (see Figure 2). As already stated before, the cut is

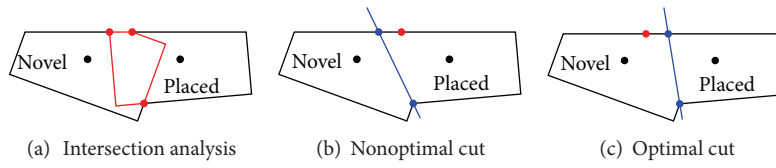


FIGURE 3: Shared cut. If both cuts satisfy the aforementioned constraints, cut (c) is chosen. In this way a smaller area is removed from the novel tile.

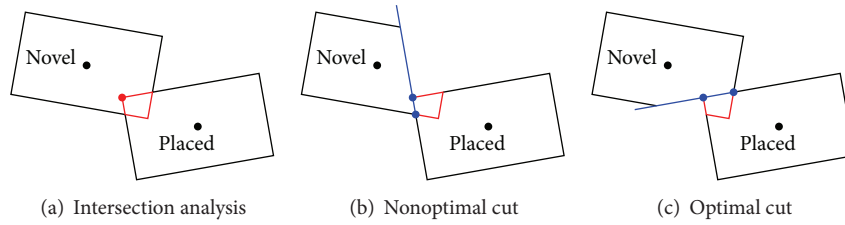


FIGURE 4: Subtractive cut. If both cuts satisfy the aforementioned constraints, cut (c) is chosen. In this way, a smaller area is removed from the novel tile.

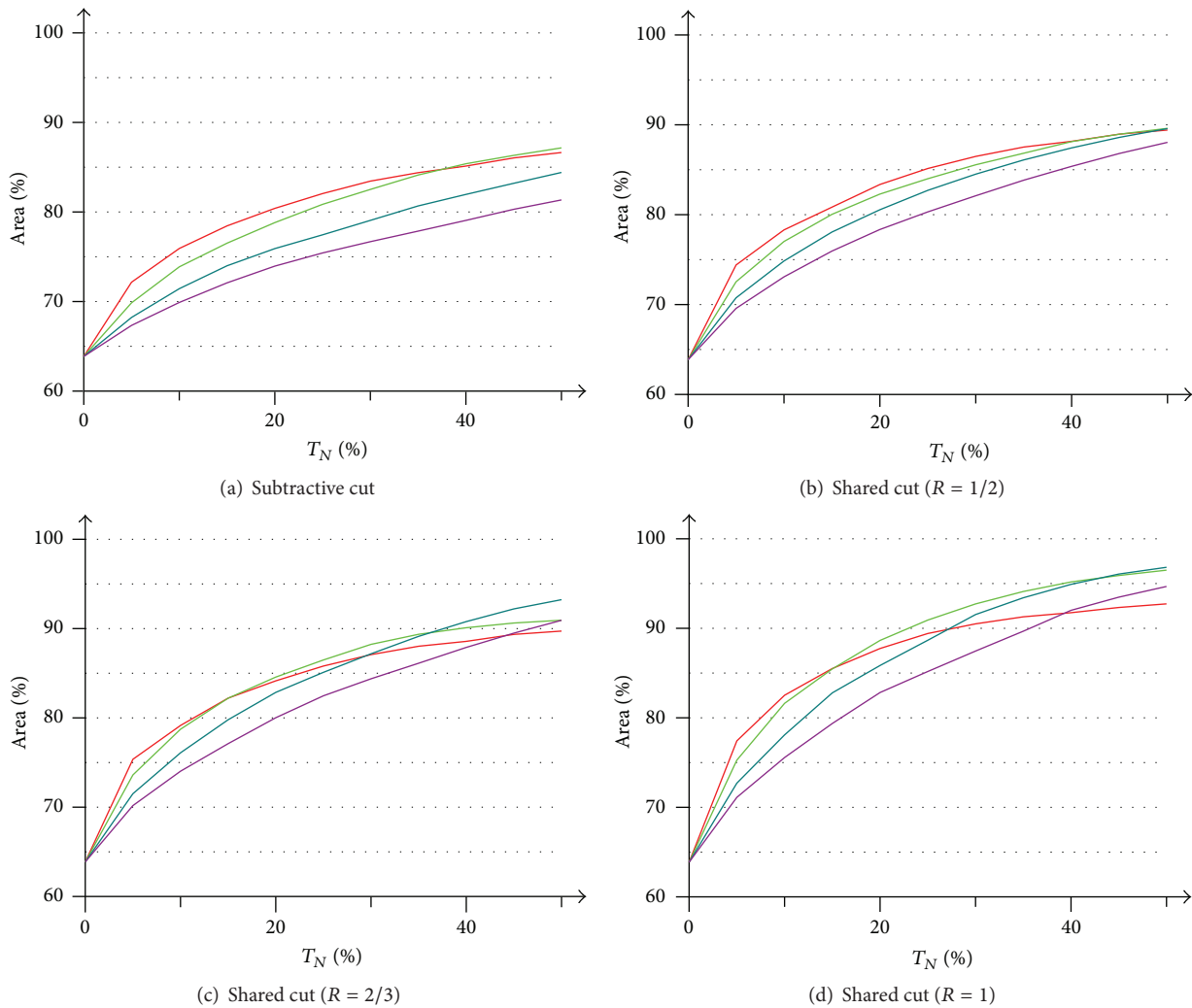


FIGURE 5: Percentage of covered area. Colors represent several R_N values: 1/4, 1/3, 1/2, and 1.

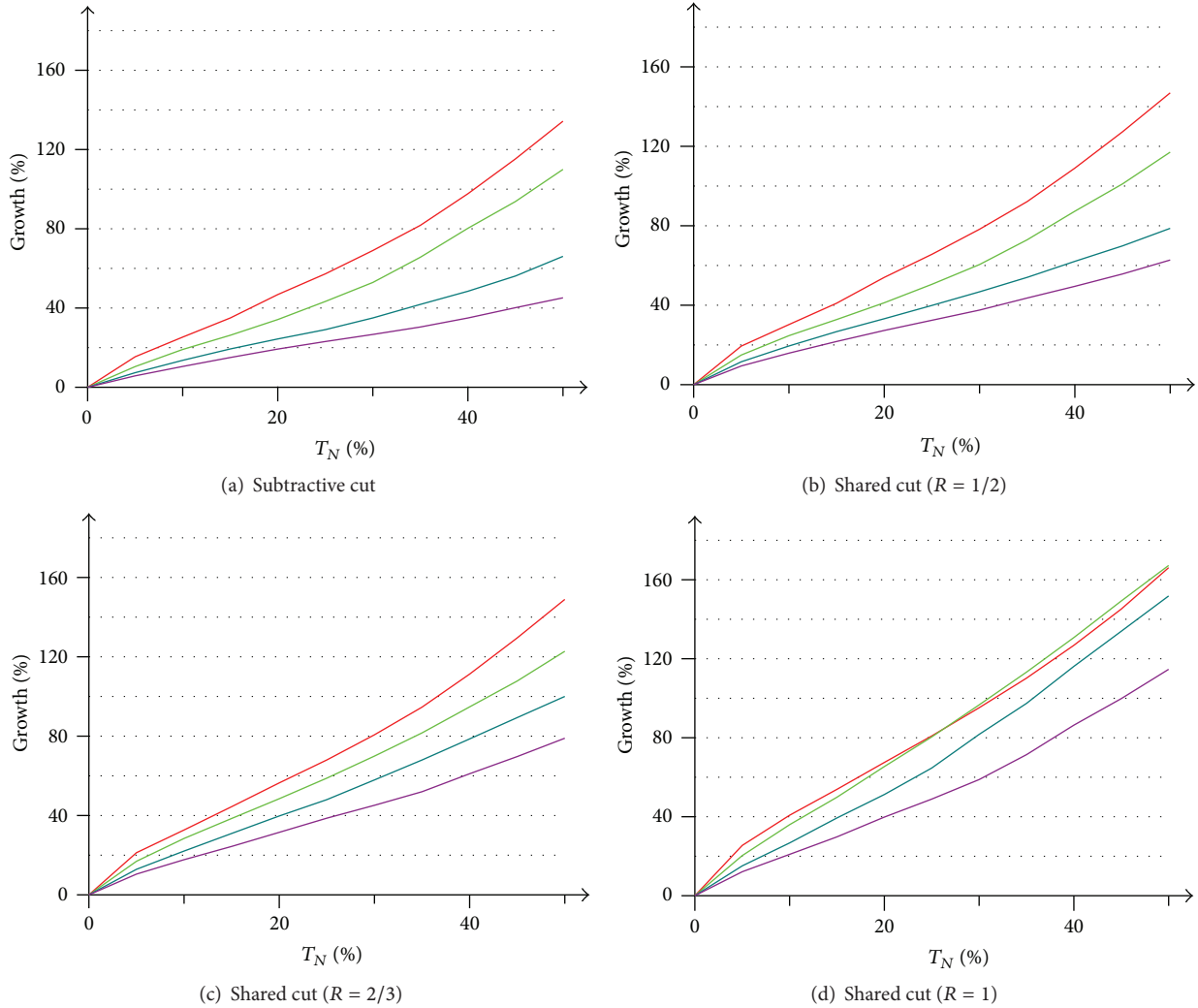


FIGURE 6: Percentage gain in the number of tiles compared to the mosaic without cuts. Colors represent several R_N values: 1/4, 1/3, 1/2, and 1.

performed both on $tile_P$ and $tile_N$, sometimes more complex configurations can arise and have to be taken into account in the tile cutting process. Considering as example, two tiles placed on the border of the canvas (Figure 3), several lines can be used for tile cutting. In these cases, the line that removes the minimum amount of the $tile_N$ is chosen. This choice increases the probability of placing the $tile_N$. In fact, this tile could undergo further cutting, and the limit imposed by the involved thresholds (see Section 4.3) could not be satisfied.

It is worth noting that the shared cut creates convex tiles without irregular parts. However, it should be carefully used because it tends to increase the sides of polygons and round shapes.

4.2. Subtractive Cut. Sometimes the shared cut cannot be used (e.g., further cutting of the placed tiles cannot be done due to the limit specified in Section 4.3); in these cases, the subtractive cut could be useful. It does not modify the already placed tiles but removes part of the novel tile.

As shown in Figure 4, two possible cuts can be considered along the side of the tile already placed. In order to preserve

more information and increase the possibility of satisfy all the constraints about tile cutting, the cut removing less area is chosen. It is worth noting that the subtractive cut gives higher importance to the already placed tiles (the orientations of their sides are taken into account in the cutting).

4.3. Tile Cutting Parameters. Both shared and subtractive tile cuts work together to achieve locally a configuration able to maintain high degree of realism in terms of comparisons with artisan works (e.g., limited the number of sides, no too complex shape, etc.) maintaining at the same time the peculiarity of an ancient mosaic (e.g., maximum coverage area, positioning along main oriented underlying structures, etc.). The two involved strategies depend on a set of thresholds detailed as follows.

- (i) T_P , maximum percentage of total cut area, from an already placed tile.
- (ii) S_P , maximum percentage of cut area, with a single cut, from an already placed tile; it should be noted that $S_P \leq T_P$.

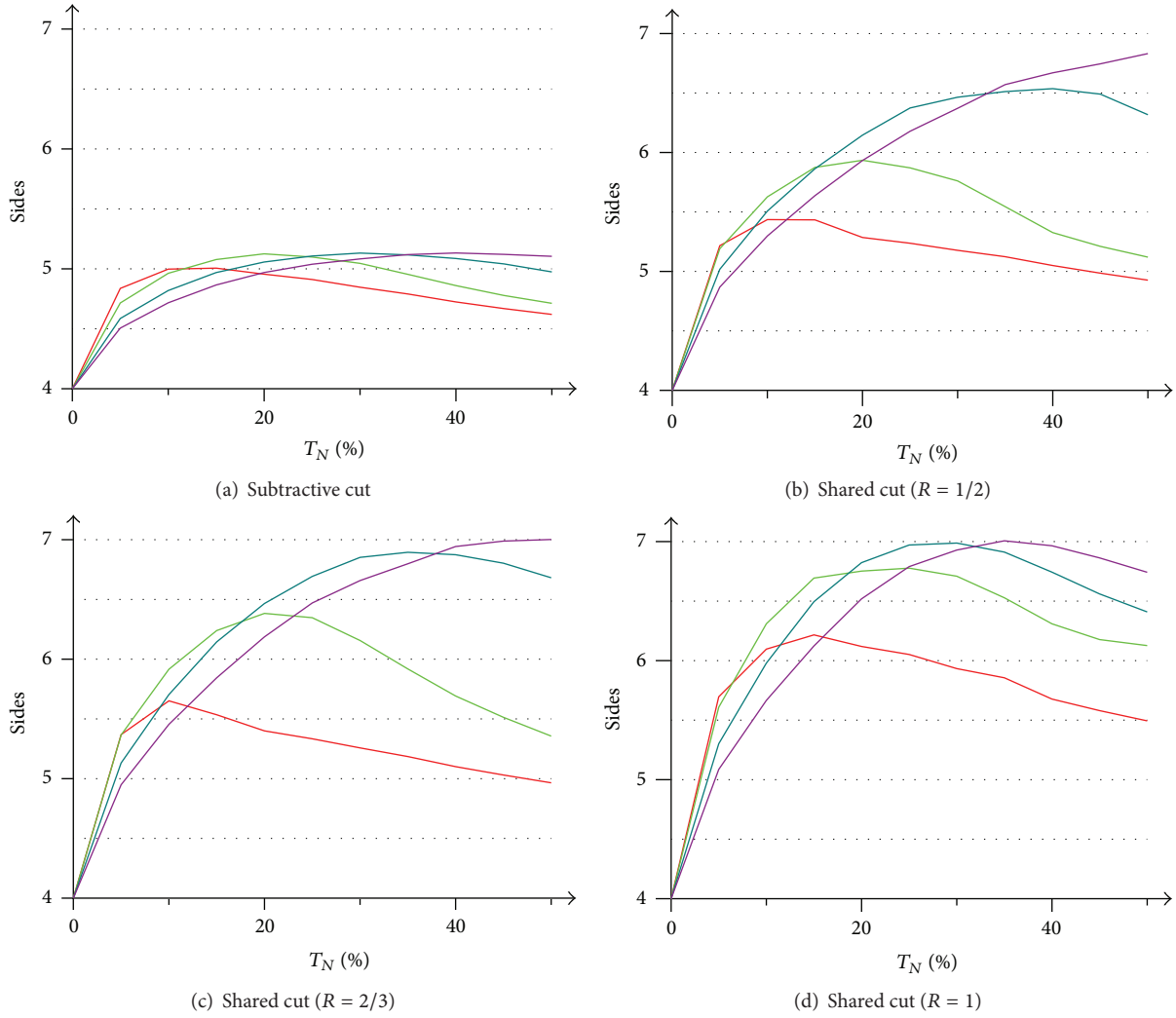


FIGURE 7: Average number of sides per tile. Colors represent several R_N values: 1/4, 1/3, 1/2, and 1.



FIGURE 8: Examples of mosaics obtained by using only subtractive cut with T_N ranging from 0% (no cut) to 50% (step of 10%) and the ratio between single and total cut R_N fixed to $1 = 2$.



FIGURE 9: Examples of mosaics obtained by using both subtractive and shared cuts with T_N ranging from 0% (no cut) to 50% (step of 10%), R_N and R fixed to $1 = 2$.



FIGURE 10: Original images used for mosaic generation: *Madonna* (1012×1351), *peasant* (910×1106), and *Kodim04* (512×768).

- (iii) T_N , maximum percentage of total cut area, from a novel tile.
- (iv) S_N , maximum percentage of cut area, with a single cut, from a novel tile; it should be noted that $S_N \leq T_N$.

Let A_0^N be the original tile area (i.e., the area of the rectangular shape the tile has when it is generated) of the novel tile and A_0^P the area of the already placed tile. Let A_i^N and A_i^P be the corresponding tile area after the i th cut.

The tile cutting of a novel tile has to satisfy the following constraints:

$$\frac{A_i^N - A_{i+1}^N}{A_0^N} \leq S_N, \quad i = 1, \dots, M_N - 1, \quad (2)$$

$$\frac{A_0^N - A_{M_N}^N}{A_0^N} \leq T_N,$$

where M_N is the overall number of cuts performed on the novel tile. The tile cutting of an already placed tile has to satisfy the following constraints:

$$\frac{A_i^P - A_{i+1}^P}{A_0^P} \leq S_P, \quad i = 1, \dots, M_P - 1, \quad (3)$$

$$\frac{A_0^P - A_{M_P}^P}{A_0^P} \leq T_P,$$

where M_P is the overall number of cuts performed on the already placed tile. Notice that there is subtractive cut if $T_P = 0$ or $S_P = 0$.

4.4. Tile Cutting Parameter Setting. To better reproduce fine details of the original picture, a good mosaic should cover the canvas as much as possible. On the other hand, to preserve the “mosaic effect,” according also to [12], the number of tiles should be limited and their shapes should be simple.

As already stated in the previous section, tile cutting depends on several parameters. In order to set them properly, some tests have been performed to study their impact on

the final generated mosaic. The variability of image content has been considered including in our test set both photographic and clip art images. Several mosaics have been generated from each image of the dataset with various tile cutting parameter values. Some objective measures have been hence derived to describe the properties of the generated mosaic. In particular we have considered the percentage of covered area (Figure 5), the percentage of gain in the number of tiles compared to the mosaic without cuts (Figure 6), and the average side number of the tiles (Figure 7). The experiments related to subtractive cut have been performed with T_N ranging from 0% (no cut) to 50% (step of 5%), and the ratio between single and total cut R_N is defined as follows:

$$R_N = \frac{S_N}{T_N}, \quad R_N \in \left\{ \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1 \right\}. \quad (4)$$

The experiments related to shared cut have been performed considering also the ratio between the total cut over the already placed and the novel tile as follows

$$R = \frac{T_P}{T_N}, \quad R \in \left\{ \frac{1}{2}, \frac{2}{3}, 1 \right\}. \quad (5)$$

The ratio between the single and total cut of the already-placed tile (R_P) is equal to R_N . The covered area is proportional to T_N and R (Figure 5). Moreover, high values of R_N (and R_P) should be avoided. Performing a big single cut (i.e., S_N and S_P close to T_N and T_P , resp.) creates some holes that cannot be covered later. As can be easily seen from Figure 6, the percentage increase in the number of tiles compared to the mosaic without cuts is proportional to T_N and R_N (and R_P).

The dependence of the average number of tile sides with respect to the aforementioned parameters has been studied in Figure 7. Decreasing R_N , small but frequent cuts are performed producing complex shapes with a higher number of sides. This trend becomes worse with shared cut at the increasing of R . Finally, side number first increases with T_N , later decreases due to the higher possibility of performing single cuts.



FIGURE 11: Visual comparisons of artificial mosaic techniques—our approach on the left (a, d, g), Liu et al. [25] in the middle (b, e, h), and Di Blasi and Gallo [22] on the right (c, f, i).

The analysis of the behavior of the proposed approach at the varying of the involved parameters has shown that the increasing of the covered area is obtained, as expected, by using a higher number of complex tiles (see Figures 8 and 9).

The overall analysis reported above has allowed us to link our computational tool with aesthetic criteria devoted to control accordingly the mosaic generation process.

5. Experimental Results

In order to show the effectiveness of the proposed method, several tests and comparisons with state-of-the-art approaches have been conducted. Specifically, a first test compares the proposed artificial mosaic generation approach with Di Blasi and Gallo [22] and Liu et al. [25] techniques both in terms of visual appearance and execution time.

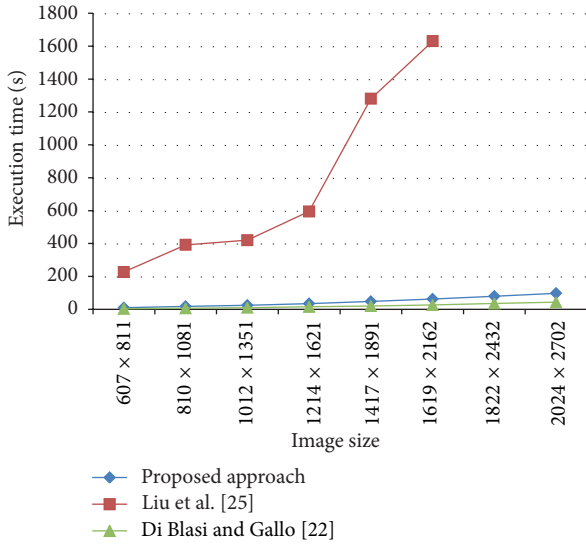


FIGURE 12: Execution time comparison at increasing image size.

TABLE 1: Execution time (seconds) of the considered approaches on the test images.

Technique	Madonna (1012×1351)	Peasant (910×1106)	Kodim04 (512×768)
Proposed approach	25 sec	20 sec	9 sec
Di Blasi and Gallo [22]	11 sec	8 sec	4 sec
Liu et al. [25]	420 sec	400 sec	155 sec

This test has been performed without considering tile cutting strategies.

Further comparisons have been also conducted to properly test the effectiveness of the tile cutting heuristics. Specifically, our solution has been compared with the technique proposed by Di Blasi and Gallo [22].

5.1. 2D Mosaic Generation Technique Comparison. To assess the overall aesthetic quality, the proposed method has been compared with the mosaics obtained by Di Blasi and Gallo [22] and Liu et al. [25] by using their original implementations. Three different images for mosaic generation (Figure 10) have been selected: *Madonna* (1012 × 1351), *Peasant* (910 × 1106), and *Kodim04* (512 × 768). For the sake of comparisons, each image has been mosaiced making use of the same rectangular shape.

As shown in Figure 11, our technique is able to preserve fine details. This happens because high-frequency areas have a higher priority on tiles placing. Moreover, gaps (i.e., area left uncovered) are properly distributed. On the contrary, Di Blasi and Gallo’s [22] algorithm creates some long “cracks” in the mosaiced images, whereas the proposed approach achieves in the same regions a pleasant smoothness. The waves generated by main edges propagate far from their original points creating unpleasant mosaic regions.

TABLE 2: Execution time (seconds) of the considered approaches on the test images employing tile cutting strategies.

Technique	Madonna (1012 × 1351)	Peasant (910 × 1106)	Kodim04 (512 × 768)
Proposed approach	70 sec	50 sec	20 sec
Di Blasi and Gallo [22]	40 sec	45 sec	20 sec

Liu et al.’s [25] approach produces very pleasant mosaics both in terms of detail preservation and gap distribution. However, it should be observed that its complexity, both in terms of memory and computational resources, is sensibly higher than that of the proposed approach (see Table 1). Moreover, this approach creates unpleasant artifacts in the left and bottom border, of the image. In Figure 12, the execution time of the considered approaches is reported at the varying of image size. The image under analysis is *Madonna* at different scales with tile size equal to 9 × 9 pixels. All the tests have been performed on a Intel(R) Core(TM)2 Duo CPU 2.53 GHz. Although our algorithm has been entirely implemented in JAVA, whereas the Liu et al.’s [25] approach is implemented in C++, we outperform it in terms of execution time. Liu et al.’s [25] technique does not scale well with the increasing of the image size; in our test, it reaches its limit with images of 1822 × 2432 pixels. The proposed approach is then able to obtain satisfactory results, similar to Liu et al. [25] and better than Di Blasi and Gallo [22] with a computational complexity considerably lower than Liu et al. [25].

5.2. Tile Cutting Experiments. In order to visually assess the quality of the proposed tile cutting heuristics, a series of mosaics have been generated. Although the optimal setting of the parameters depends on the image under analysis and on the subjectivity of the user, a good tradeoff is as the following:

$$R = \frac{1}{2}, \quad R_N = \frac{1}{2}, \quad T_N = 35\%, \quad S_N = 17.5\% \quad (6)$$

$$T_P = 17.5\%, \quad S_P = 8.75\%.$$

To further validate our approach, some comparisons have been performed considering the approach proposed in [22] (see Figures 13 and 14). Five images have been used in our tests: *Madonna*, *Peasant*, *Kodim04*, *the Scream*, and *Kodim09*.

Di Blasi and Gallo [22] obtain a high degree of realism. The percentage of covered area is pretty high but it is achieved by making use of an elevated number of small and complex tiles. Moreover, the algorithm is not able to properly combine information coming from different edges of the images, producing some unpleasant artifacts. On the contrary, our approach obtains a satisfactory coverage area maintaining at the same time an acceptable number of tiles with a low number of sides. Moreover, the properties of GVF about edge information propagation permit us to obtain graceful results (no visually artefact). It should be noted that a simple 2D mosaic can be better used as starting point for other image manipulations (e.g., 3D mosaic generation [27]).



FIGURE 13: Visual comparisons of artificial mosaic techniques—our approach on the left (a, c, e, g, i, k) and Di Blasi and Gallo [22] on the right (b, d, f, h, j, l).

The execution time related to the considered techniques has been also reported in Table 2.

6. Conclusions

In this paper, we have proposed a novel algorithm for artificial mosaic generation. Our contribution has been twofold. Several comparisons have been performed among several

state-of-the-art approaches both in terms of aesthetical appearance and computational demands. Moreover, several heuristics have been introduced to properly manage tile cutting, producing hence graceful mosaics with irregular tiles. The effectiveness of our approach has been demonstrated through a series of experiments and comparisons with recent techniques. Future work will be devoted to study the dependence of tile size with respect to image content and size.

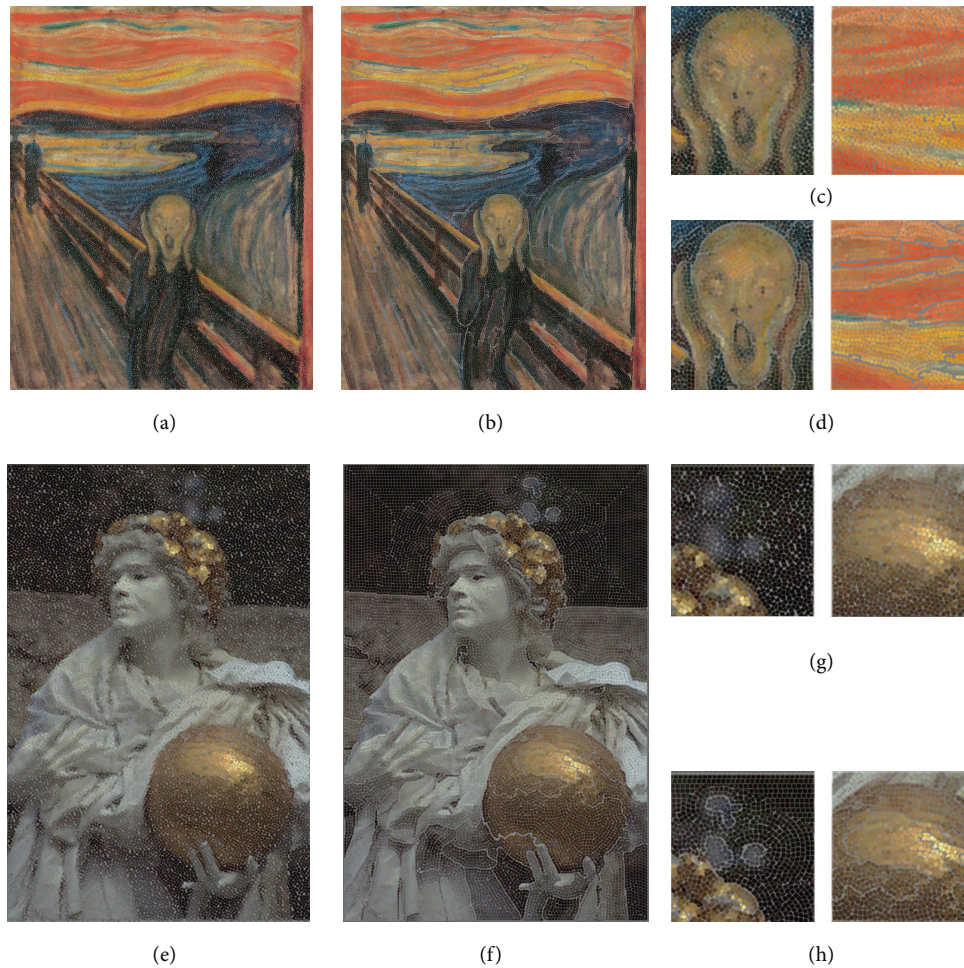


FIGURE 14: Visual comparisons of artificial mosaic techniques—our approach on the left (a, c, e, g) and Di Blasi and Gallo [22] on the right (b, d, f, h).

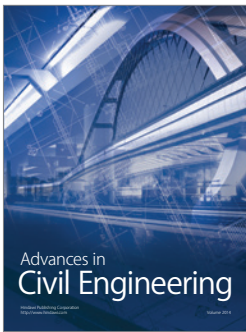
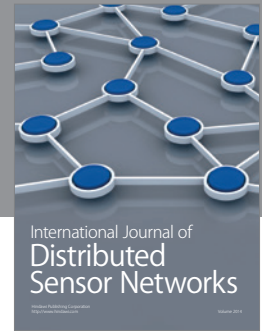
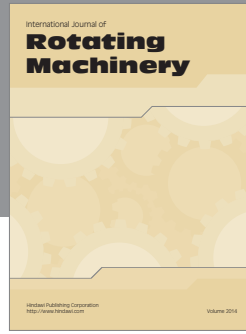
Acknowledgment

This work was supported by Regione Sicilia PO/FESR 4.1.1.2 under the EMOCUBE Project.

References

- [1] J. Collomosse, *Higher level techniques for the artistic rendering of images and video [Ph.D. thesis]*, University of Bath, UK, 2004.
- [2] T. Saito and T. Takahashi, "Comprehensible rendering of 3-d shapes," in *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '90)*, pp. 197–206, 1990.
- [3] A. A. Gooch, "Towards mapping the field of non-photorealistic rendering," in *Proceedings of the 8th Meeting of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR '10)*, pp. 159–164, June 2010.
- [4] A. Hertzmann, "Non-photorealistic rendering and the science of art," in *Proceedings of the 8th Meeting of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR '10)*, pp. 147–157, June 2010.
- [5] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the Computer Graphics Annual Conference (SIGGRAPH '01)*, pp. 327–340, August 2001.
- [6] J. P. Collomosse and P. M. Hall, "Cubist style rendering from photographs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 4, pp. 443–453, 2003.
- [7] C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischer, and D. H. Salesin, "Computer-generated watercolor," in *Proceedings of the Conference on Computer Graphics (SIGGRAPH '97)*, pp. 421–430, August 1997.
- [8] L. Neumann, M. Sbert, B. Gooch, and W. Purgathofer, "Computational aesthetics 2005," in *Proceedings of the Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, 2005.
- [9] S. Battiato, G. Di Blasi, G. M. Farinella, and G. Gallo, "Digital mosaic frameworks—an overview," *Computer Graphics Forum*, vol. 26, no. 4, pp. 794–812, 2007.
- [10] S. Battiato, G. Di Blasi, G. Gallo, and G. Puglisi, "Digital reproduction of ancient mosaics," in *Digital Imaging for Cultural Heritage Preservation: Analysis, Restoration and Reconstruction of Ancient Artworks*, S. Battiato, G. Gallo, and F. Stanco, Eds., CRC Press, 2011.
- [11] M. Du Sautoy, *Symmetry: A Journey into the Patterns of Nature*, Harper Collins, 2008.

- [12] I. Fiorentini Roncuzzi and E. Fiorentini, *Mosaic: Materials, Techniques and History*, MWeV, 2002.
- [13] A. Hausner, "Simulating decorative mosaics," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 573–580, 2001.
- [14] G. Elber and G. Wolberg, "Rendering traditional mosaics," *Visual Computer*, vol. 19, no. 1, pp. 67–78, 2003.
- [15] L. Fritzsche, H. Hellwig, S. Hiller, and O. Deussen, "Interactive design of authentic looking mosaics using voronoi structures," in *Proceedings of the 2nd International Symposium on Voronoi Diagrams in Science and Engineering (VD '05)*, 2005.
- [16] Y. Liu, O. Veksler, and O. Juan, "Simulating classic mosaics with graph cuts," in *Proceedings of Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 55–70, 2007.
- [17] S. Battiato, G. Di Blasi, G. Gallo, G. C. Guarnera, and G. Puglisi, "Artificial mosaics by gradient vector flow," *Proceedings of EUROGRAPHICS*, 2008.
- [18] S. Battiato, G. Di Blasi, G. Gallo, G. C. Guarnera, and G. Puglisi, "A novel artificial mosaic generation technique driven by local gradient analysis," in *Proceedings of the Computational Science (ICCS '08)*, vol. 5102 of *Lecture Notes in Computer Science*, pp. 76–85, 2008.
- [19] H. Li and D. Mould, "Artistic tessellations by growing curves," in *Proceedings of the 9th International Symposium on Non-Photorealistic Animation and Rendering (NPAR '11)*, pp. 125–134, 2011.
- [20] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [21] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.
- [22] G. Di Blasi and G. Gallo, "Artificial mosaics," *Visual Computer*, vol. 21, no. 6, pp. 373–383, 2005.
- [23] S. Battiato, G. Di Blasi, G. 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 '06)*, pp. 133–140, 2006.
- [24] S. Schlechtweg, T. Germer, and T. Strothotte, "RenderBots—multi-agent systems for direct image generation," *Computer Graphics Forum*, vol. 24, no. 2, pp. 137–148, 2005.
- [25] Y. Liu, O. Veksler, and O. Juan, "Generating classic mosaics with graph cuts," *Computer Graphics Forum*, vol. 29, no. 8, pp. 2387–2399, 2010.
- [26] L. G. Roberts, *Machine Perception of Three-Dimensional Solids, Outstanding Dissertations in the Computer Sciences*, Garland Publishing, New York, NY, USA, 1963.
- [27] S. Battiato and G. Puglisi, "3D ancient mosaics," in *Proceedings of the 18th ACM International Conference on Multimedia ACM Multimedia (MM '10)*, pp. 1571–1574, October 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

