Artificial Mosaics by Gradient Vector Flow

S. Battiato[1], G. Di Blasi[1], G. Gallo[1], G. C. Guarnera[1] and G. Puglisi[1]

[1] Dipartimento di Matematica e Informatica, University of Catania, Italy

Abstract

In this paper a novel approach for artificial mosaic generation is proposed. Gradient Vector Flow computation together with heuristics to maximise the covered mosaic area are used. The high frequency details are managed in a global way allowing to preserve the mosaic-style also for small ones. Experiments and comparisons with previous works confirm the effectiveness of the proposed algorithm.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

1. Introduction

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. Tile dataset, constraints on positioning, deformations and rotations of the tiles are very influential upon the final results. The creation of a digital mosaic resembling the visual style of a traditional one is a challenging problem because it has to take into account the polygonal shape of the tiles, the size of the tiles, the need to pack the tiles as densely as possible. Moreover tile orientation has a strong visual influence on the overall perception of the mosaic. Orientation cannot be arbitrary but it is constrained to follow the gestalt choices of the source picture.

The translation of a raster source image into a digital mosaic may take the form of a mathematical optimization problem [BDFG07]. In this perspective the creation of images looking like ancient mosaics can be formalized as follows:

Given an image I in the plane R^2 and a vector field $\Phi(x,y)$ defined on that region by the influence of the edges of I, find N sites $P_i(x_i,y_i)$ in I and place N rectangles, one at each P_i, oriented with sides parallel to $\Phi(x_i,y_i)$, such that all rectangles are disjoint, the area they cover is maximized and each tile is colored by a color which reproduces the image portion covered by the tile [Hau01].

The first attempt to reproduce a realistic ancient mosaic was presented by Hausner [Hau01]. He obtained very results using Centroidal Voronoi Diagrams (CVD), user selected edge features, $L1$ (Manhattan) distance and graphic hardware acceleration. The number of iterations to reach convergence is one of the main drawback of this technique, expecially when there is no direct access to a graphic accelerator. Another algorithm for the creation of ancient mosaics is presented in [DG05] and in [BDFG06]. This approach, based on directional guidelines and distance transform, leads to very realistic results. The overall asymptotic complexity of this algorithm is linear with respect to the image size. In [SGS05] an approach for stroke-based rendering based on multi-agent systems is presented. The authors name the agents RenderBots. Those are individual agents representing one stroke. Recently an approach based on graph-cut optimization algorithms has been presented [LVJ07]. It is able to work on tile positioning without an explicit edge detection phase. A rather complete discussion with visual comparisons about the existing methods in the field of artificial mosaic can be found in [BDFG07].

In this paper we adopt a "one-after-one" tile positioning and orienting strategy as in [DG05] and in [BDFG06]. The proposed approach uses Gradient Vector Flow (GVF) [XP98]. GVF is able to capture the overall gradient behaviour in a proper way. The main novelty introduced in this paper is a new way to deal with high-frequency details. Almost all previous approaches filtered-out such components to simplify the tile positioning heuristics. We propose to use GVF of a raster image to guide the positioning of the tiles. GVF computation simplifies tile positioning, because the generation process of the flow is constrained to be smooth, maintaining at the same time the information of the small details. Another novelty is the use of heuristics to maximize the image area filled with tiles. The results obtained with the new technique are of higher visual quality than the results obtained with other state of the art algorithms [BDFG07].

The paper is structured as follows. Next section describes in detail the proposed methodology. Experiments are reported in Section 3 whereas Section 4 closes the paper tracking direction for future works.

2. Proposed Algorithm

To emulate the style of an ancient artisan we proposed a technique based on the following two steps:

- GVF field computation based on [XP98] algorithm;
- rule based tile positioning.

GVF is a dense force field presented in [XP98] in order to solve the classical problems affecting snakes (sensitivity to initialization and poor convergence to boundary concavity). Starting from the gradient of an image, this force field is computed through diffusion equations.

GVF is a field of vectors $\mathbf{v} = [v, u]$ that minimizes the following energy function:

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

where the subscripts represent partial derivates along x and y axes respectively, μ is a regularization parameter and $|\nabla f|$ is the gradient magnitude computed from the input image. GVF field values are close to $|\nabla f|$ where this quantity is large (energy E, to be minimized, is dominated by $|\nabla f|^2 |\mathbf{v} - \nabla f|^2$). GVF field values 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.

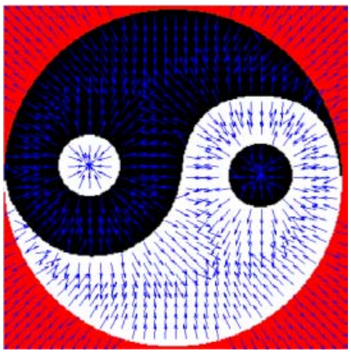

Figure 1: *Input image and its corresponding GVF field.*

The GVF force field can be used to effectively drive tiles positioning. Edge information is preserved, it is propagated in the close regions and merged together in a smooth way.

Let I be the input image to be mosaicized. 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 result is computed. The equalization process, especially for natural images, allows to normalize the overall gradient distribution. The horizontal and vertical components of the gradient $\nabla L(I)$ are used as input for the GVF algorithm. Notice that in the actual implementation gradient computation is performed using Robert's Kernel.

All the tiles have the same rectangular shape and the same size. We ask that tiles 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 $|GVF(I)|$. We impose that the tile orientation is equal to the GVF direction in its central point. As a consequence the areas near to the more relevant edges of the input images are taken care first.

The second step of the algorithm is devoted to cover the homogeneous regions of the image. This is accomplished simply placing each tile one by one left to right, up to bottom, starting from the upper-left corner of the image. This technique leads to aesthetically pleasant results, by preserving the orientation of the main edges and covering a wide portion of pixels with densely packed tiles. The algorithm can be summarized as follows:

```
Input: a raster image $I$
$L(I) \leftarrow$ Luminance($I$)
$G(I) \leftarrow$ Robert's Gradient(Equalize($L(I)$))
$[u,v] \leftarrow GVF(|G(I)|_{\infty}, \mu, nIterations)$
$gvf(I) \leftarrow (u^2+v^2)^{1/2}$
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
  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
    Skip tile positioning.
for $j \leftarrow 1$ to length($I$)
  for $i \leftarrow 1$ to width($I$)
    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
      Skip tile positioning.
```

3. Experimental Results

The above algorithm has been implemented in JAVA, using an external MATLAB module for GVF computation [GVF]. The method as been compared, to assess the aesthetic quality, with the mosaics obtained by [DG05] and [LVJ07] by using their original implementations. We wish to thank all the authors of these papers for providing the output of their

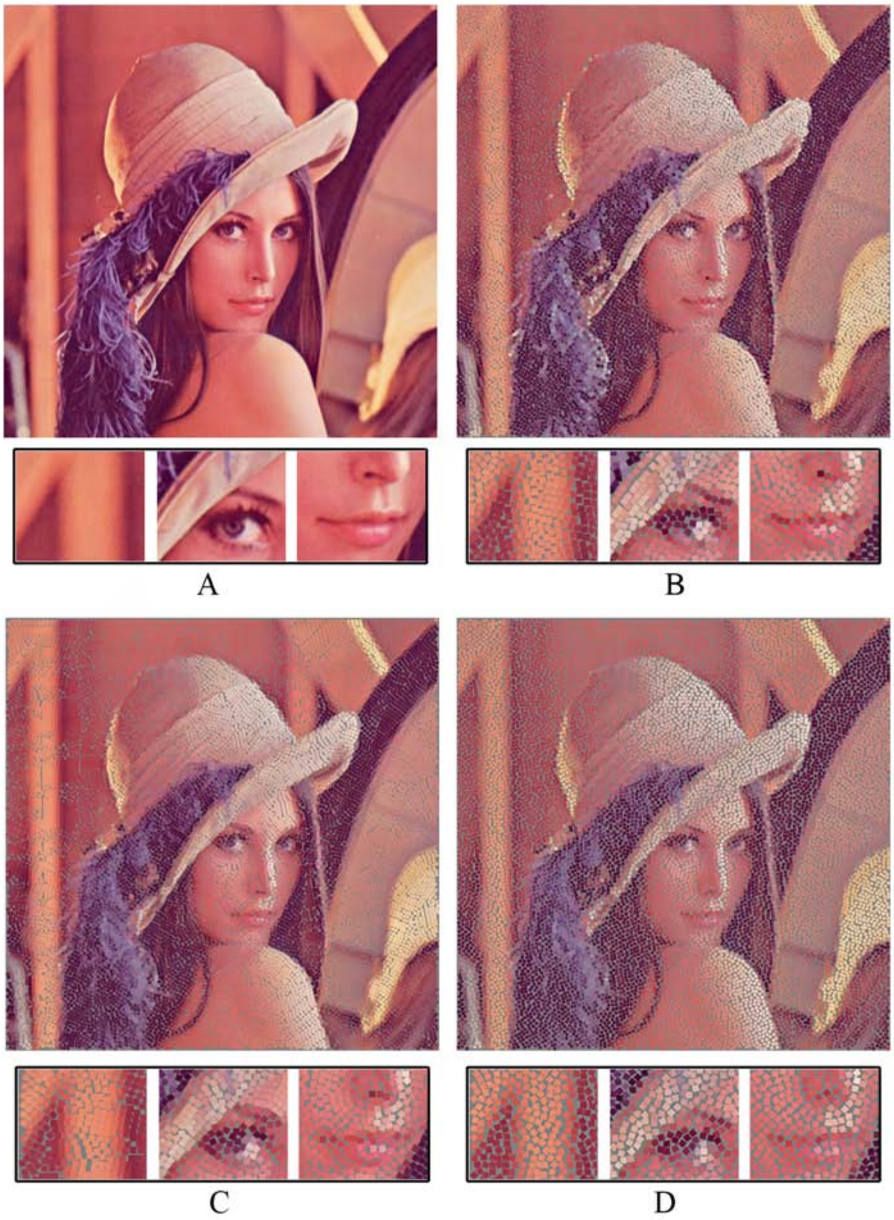

Figure 2: *Visual comparison between mosaics generated by our approach (B), [DG05] (C) and [LVJ07] (D), applied on input image (A) considering in all cases the same tile size* 5×5*.*

techniques upon our request. For sake of comparison Figure 2 reports the mosaic obtained from the standard Lena picture. A first advantage of the novel technique is that it is able to better preserve fine details. This happens because high frequency areas are prioritarily filled. Observe, to support this claim, the areas around Lena's nose, lips and high brow. Another performance index of mosaic algorithms is the amount and spatial distribution of untiled space. The area left uncovered by the proposed technique is comparable with the amount of uncovered area left by [DG05], but gaps are here better distributed. For example the constraints of [DG05] force the appearance of a long "crack" in the vertical band on the wall behind Lena, whereas the proposed approach has in the same region a pleasant smoothness. Relatively to [LVJ07] the uncovered area left by our technique is considerably less (see Table 1). Observe that a higher percentage of covered area leads to a better preservation of the original colors of the same picture. The perceived texture obtained with the proposed technique appears, finally, less chaotic than the texture obtained with [LVJ07]. The parameters adopted to produce Figure 2 are: tile size 5×5, image size 667×667.

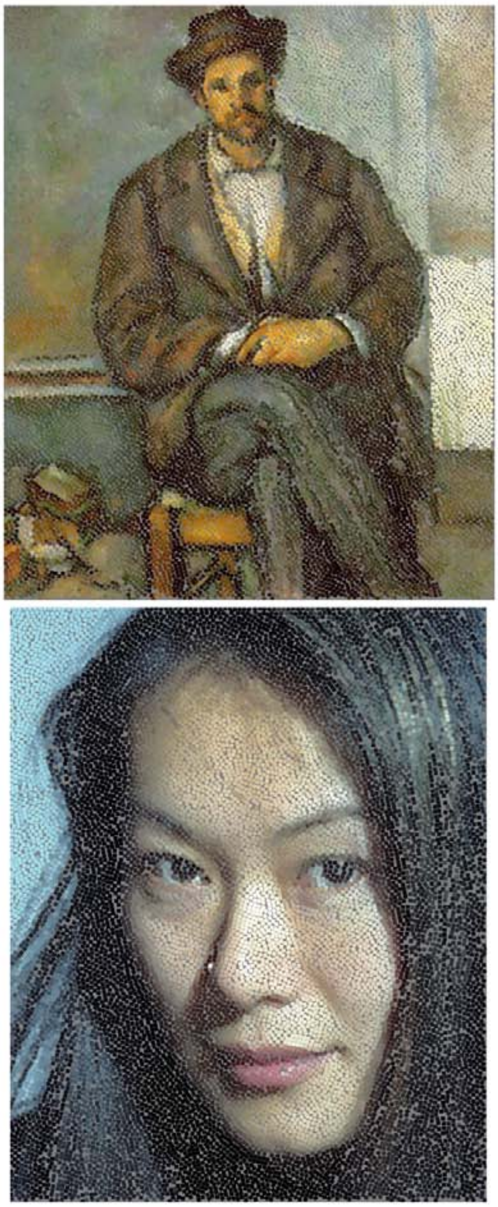

Figure 3: *Some mosaics generated by our approach.*

The overall complexity of the proposed technique is $O(kn) + O(nlogn)$, where n is the number of pixels in the source image. To provide information about typical processing time we report that the Lena picture (size 667x677) has been processed taking 77.6 sec to obtain the GVF and 2.3 sec for the tesselation. The processing has been performed on a AMD Opteron 150, 2.4 Ghz. Please notice tha GVF computation is done using a separate MATLAB module [GVF] while tile positioning is a JAVA module. Further results are shown in Figure 3. The mosaicized images can be also downloaded at the following web address:

`http://svg.dmi.unict.it/iplab/download/.`

Technique	Number of tiles	Covered area
Our Method	13514	75.9%
[DG05]	13994	78.6%
[LVJ07]	11115	62.5%

Table 1: *Number of tiles and covered area comparison between various approaches.*

4. Conclusions

We propose a novel technique to produce a traditionally looking mosaic from a digital source picture. The new technique overcomes explicit edge detection using the Gradient Vector Flow. Tests show that the new technique produces aesthetically pleasant images that have a greater fidelity in dealing with fine details and a better management of gaps. The proposed technique does not cut tiles. Indeed the next research step will integrate tile cutting. The use of other GVF methos to drive tile positioning will be also explored.

References

[BDFG06] BATTIATO S., DI BLASI G., FARINELLA G. M., GALLO G.: A novel technique for opus vermiculatum mosaic rendering. In *Proc. ACM/WSCG2006* (2006).

[BDFG07] BATTIATO S., DI BLASI G., FARINELLA G. M., GALLO G.: Digital mosaic frameworks - an overview. *Computer Graphics Forum 26*, 4 (2007).

[DG05] DI BLASI G., GALLO G.: Artificial mosaics. *The Visual Computer 21*, 6 (2005), 373–383.

[GVF] `http://www.iacl.ece.jhu.edu/resources/.`

[Hau01] HAUSNER A.: Simulating decorative mosaics. In *Proc. SIGGRAPH 2001* (2001), pp. 573–580.

[LVJ07] LIU Y., VEKSLER O., JUAN O.: Simulating classic mosaics with graph cuts. In *Proc. EMMCVPR* (2007), pp. 55–70.

[SGS05] SCHLECHTWEG S., GERMER T., STROTHOTTE T.: Renderbots — multi-agent systems for direct image generation. *Computer Graphics Forum 24*, 2 (2005), 137–148.

[XP98] XU C., PRINCE L.: Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing 7*, 3 (1998), 359–369.