A Novel Technique for Opus Vermiculatum Mosaic Rendering

S. Battiato G. Di Blasi G.M. Farinella G. Gallo

IPLab – Image Processing Laboratory
http://www.dmi.unict.it/~iplab/
Dipartimento di Matematica e Informatica
University of Catania, Via Andrea Doria 6 – 95125, Catania (Italy

{battiato, gdiblasi, gfarinella, gallo}@dmi.unict.it

ABSTRACT

In this paper we present a method to generate a digital mosaic starting from a raster input image. Mosaics generation of artistic quality is challenging. The basic elements, the tiles, typically small polygons, must be packed tightly, emphasizing orientations chosen by the artist. An ad-hoc boundaries detection have to be performed according to the directional guidelines. Different mosaic styles can be automatically rendered, depending on artistic techniques considered ("opus musivum", "opus vermiculatum", etc.).
The proposed method is able to reproduce the colors of the original image emphasizing relevant boundaries by placing tiles along their direction. The boundaries detection is based on the statistical region merging algorithm. In particular the technique is able to reproduce the "opus vermiculatum" mosaic style.
Several examples reported in the paper show how the right mixture of mathematical tools together with century proved ideas from mosaicists may lead to impressive results.

Keywords
mosaic, non-photorealistic rendering, distance transform, image processing and enhancement, statistical region merging.

1. INTRODUCTION

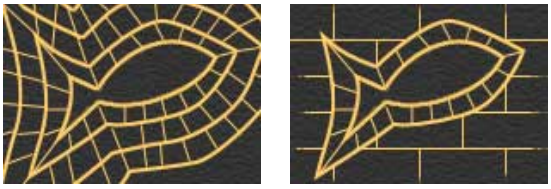
Mosaics are artworks constituted by cementing together small colored tiles. Probably, they are the first example of image synthesis techniques based on discrete primitives. A smart and judicious use of orientation, shape and size may allow to convey much more information than the uniform or random distribution of *N* graphic primitives (like pixels, dots, etc.). For example, ancient mosaicists avoided lining up their tiles in rectangular grids, because such grids emphasize only horizontal and vertical lines. Such artifacts may distract the observer from seeing the overall picture described. To overcame such potential drawback, old masters placed tiles emphasizing the strong edges of the main subject to be represented, as shown in Figure 1.

Conference proceedings ISBN 80-86943-03-8
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

Figure 1. Example of mosaic.

In [Dib05a] a technique able to render traditional looking mosaics is presented. As the authors point out, the main problem to solve in the mosaic rendering is the directional guidelines detection. Directional guidelines are related with the salient edges of the image and the unsupervised detection is, "per se", an interesting and challenging problem. Moreover, directional guidelines and edges are two related but different features. The use of classical edge detector algorithms (for example [Mee01]) cannot hence be transferred without great care to the problem at hand.
The authors present also a simple technique to detect the directional guidelines, but they grant that although they have adopted a practical solution that is satisfactory for the application, the problem deserves a

theoretical and algorithmic deeper investigation.
In this paper we present another directional guidelines detection technique based on the Gestalt movement idea of the perceptual grouping [Kof22]. In particular, the statistical region merging algorithm [Noc04] has been used. This method allows a more precise detection of the most important feature curves of an image, taking into account simple statistical consideration in a pixel neighborhood.
We extend the technique presented in [Dib05a] in order to produce another mosaic style called "opus vermiculatum" (see Figure 2).



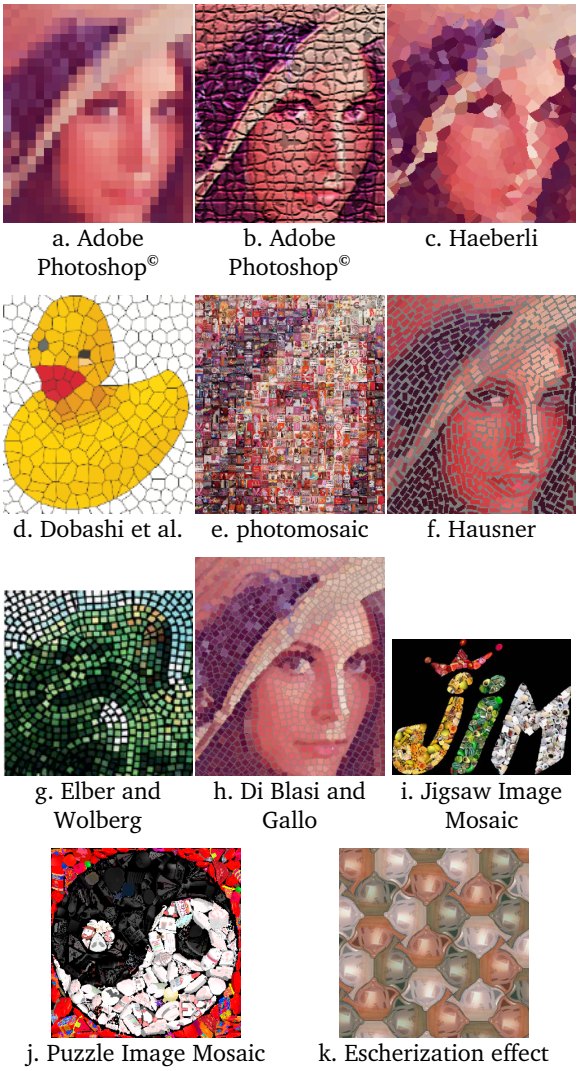
a. Opus musivum b. Opus vermiculatum

Figure 2. Examples of ancient mosaics stiles.

As the epithet of "opus musivum" says this means of "quality worth of the muses", of great visual refinery and effect. "Opus vermiculatum" takes its name from the Latin for "worm". It refers to lines of tiles that snake around a feature in the mosaic. Often two or three rows of "opus vermiculatum" appear like a halo around something in a mosaic picture, helping it stand out from the background (see Figure 1).
The rendering of "opus vermiculatum" mosaics requires a clear separation between foreground and background because the two regions of the image have to be managed in different ways. The foreground region is covered as an "opus musivum" (see [Dib05a] for details), while the background region have to be covered by a regular grid of tiles (eventually perturbed by a random noise in size, position and rotation).
The rest of the paper is organized as follows: in Section 2 we summarize previous related works, Section 3 explains the details of the proposed algorithm. In Section 4 some experimental results are reported. Finally in Section 5 we suggest directions for future works and research.

2. REVIEW OF PREVIOUS WORKS

Computer Graphics attempts to simulate mosaics inscribe themselves into the broader area of non-photorealistic rendering (NPR).
Although mosaics are a traditional art form attempts to simulate them in the digital realm are recent. Commercial image processing software provide "mosaic filters" to obtain tessellated images (the examples in Figure 3a and Figure 3b have been produced with Adobe Photoshop© [Ado06]).
More sophisticated approaches try to adopt smart strategies using computational geometry together with image processing tricks.
Haeberli [Hae90] used Voronoi diagrams, placing the sites at random and filling each region with a color sampled from the image. This approach tessellates the image, but tile shapes are too variable and do not attempt to follow edge features (see Figure 3c). This technique is also available in many user-end applications under the name of "crystallization" and it simulates the typical effect of some glass windows in the churches. In [Dob02] Dobashi et al. reprised the Haeberli's idea obtaining good results (see Figure 3d). They use an optimization technique based on the edge features of the image which avoids to place a tile across a meaningful border of the picture.



a. Adobe Photoshop© b. Adobe Photoshop© c. Haeberli
d. Dobashi et al. e. photomosaic f. Hausner
g. Elber and Wolberg h. Di Blasi and Gallo i. Jigsaw Image Mosaic
j. Puzzle Image Mosaic k. Escherization effect

Figure 3. Mosaic effects.

"Photomosaic" [Sil97] transforms an input image into a rectangular grid of thumbnail images (see Figure 3e). In this approach the algorithm searches in a large database of images to find the best approximation of a block of pixels in the main image. The resulting effect is very impressive, but even in this case no edge features are respected. The idea was successively extended by Klein et al. [Kle02] to videos obtaining a video mosaic. Recently Di Blasi and Petralia [Dib05b] presented an approach to speed up the search process

based on the Antipole strategy [Can05]; the same technique has been previously used for fast texture synthesis [Bat03] and fast colorization of gray images [Dib03].
Hausner [Hau01] obtains very good results using centroidal Voronoi diagrams, edge features, L_1 (Manhattan) distance and graphic hardware acceleration to optimize the results (Figure 3f). A very advanced approach to the rendering of traditional mosaics is presented in [Elb03]. This technique is based on offset curves that get trimmed-off the self intersecting segments with the guidance of Voronoi diagrams. The algorithm requires a mathematical description of the edges (e.g. B-splines) allowing an accurate tile placement (Figure 3g). Other advantages of this approach is the use of variable size tiles. Although the results are very good the technique seems limited to the case of a single, user-selected and close edge curve. As previously mentioned another approach for the generation of ancient mosaics is presented in [Dib05a]; this approach is based on directional guidelines, distance transform, mathematical tools and century proved ideas from mosaicists leading out impressive results (Figure 3h).
Kim and Pellacini [Kim02] introduce a mosaicing technique where image tiles of arbitrary shapes are used to compose the final picture. The idea is quite similar to the photomosaic, but the final effect is very different and interesting (Figure 3i). Another approach for the creation of the same kind of mosaics is presented in [Dib05c]; the described approach leads to impressive results in an acceptable computation time (Figure 3j).
For sake of completeness we also cite "Escherization" [Kap00], a technique that produces tilings of the plane using slightly distorted version of an image (Figure 3k). It relies on symmetry groups and regular tilings. It is very different from the other kind of methods we reviewed above and it is aimed to the production of a sophisticated kind of aesthetic effects different than mosaics.

3. THE PROPOSED ALGORITHM

In this Section we present our mosaic algorithm describing the segmentation algorithm used to distinguish interactively foreground and background areas together with the details needed to obtain the "opus vermiculatum" mosaics.

3.1 Statistical Perceptual Grouping Segmentation for Guidelines Detection

Segmentation is a process useful to subdivide an image into its meaningful regions, corresponding to objects represented in a scene. An image segmentation allows an easy guidelines detection: boundaries of each detected region are considered as guidelines. The precision of the segmentation is a decisive factor for a good guidelines detection.
Gestalt movement [Kof22] claims that the perceptual grouping has a fundamental role in human perception. Following this idea, the visual ability is reformulated as a statistical inference problem.
Statistical Region Merging [Noc04] (SRM) is a recent segmentation technique able to capture the main structural components of a digital image using a simple but effective statistical analysis.
SRM is based on two components:

1. *Statistical Merging Predicate*, used to establish if two regions have to be merged;
2. *Merging Order*, used to establish the order to test the Statistical Merging Predicate;

The pseudo-code of the algorithm is reported below:

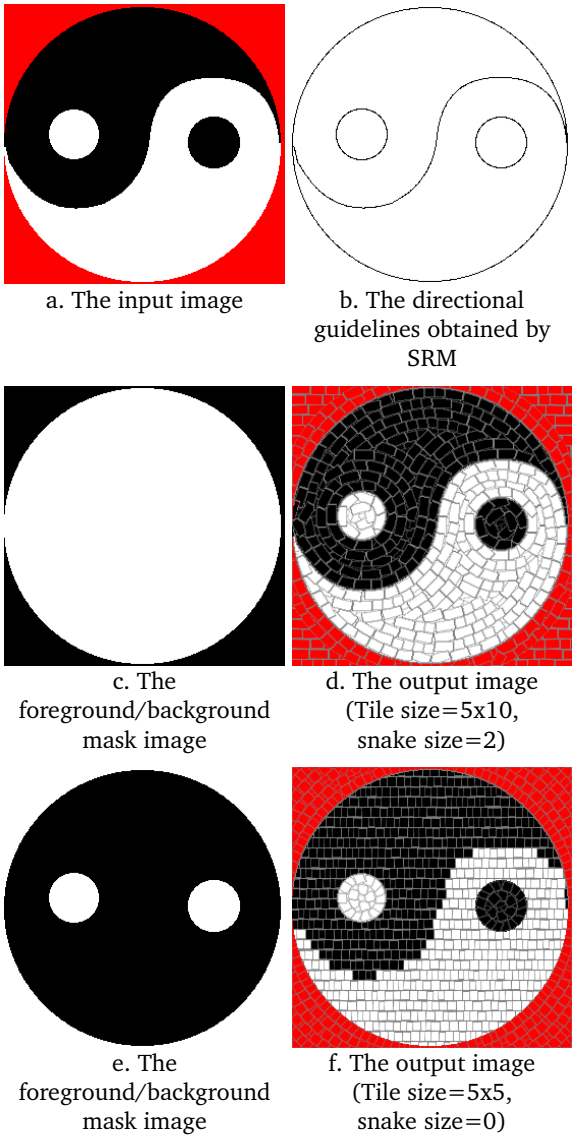
```
Input: an image I

Let S_I be the set of the 4-connexity
couples of adjacent pixel in I.

S'_I=Order_increasing(S_I, f);

For i=1 to |S'_I| do

  If R(p_i)≠R(p'_i) and P(R(p_i),R(p'_i)) = true
     then Union(R(p_i), R(p'_i))

end For
```

$f(p, p')$ is a real-valued function, with p and p' adjacent (4-connexity) pixels in I. f is used to establish the Merging Order (i.e. $Order\_increasing(S_I, f)$). The function f used in our case approximates the following invariant: "when any test between two true regions occurs, that means that all tests inside each of the two true regions have previously occurred".
An important tuning parameter of SRM is the merging factor; it allows to control the coarseness of the segmentation. In our case we use a high merging factor value to force an explicit over-segmentation needed for our purposes (as described in Section 3.2).
It is straightforward to verify that the time asymptotic complexity is linear in the number of pixels of the image.
SRM has been recently successfully used for microarray analysis ([Bat06b]). Experimental results show that SRM can be effectively used to solve the guidelines detection problem.

3.2 Rendering of Opus Vermiculatum Mosaics

In order to produce an "opus vermiculatum" mosaic, an "a priori" segmentation of the input images is needed: as previously stated such artistic mosaic is characterized by the different styles used to place tiles in the foreground and in the background. In the foreground, tiles are placed as in the "opus musivum" mosaic, while in the background tiles have to be placed in a regular grid, eventually perturbed by a random noise in size, position and rotation.
Moreover, foreground region have to be partially expanded in order to simulate the lines of tiles that snake around the foreground features (often two or three rows).
It could be possible to use some algorithms able to automatically find foreground and background regions

in an image (see for example [Kim05] and [Pic04]): these methods are often imprecise and noise dependent giving out, for our purposes, poor and unuseful information. Moreover, we need also a description of the details inside the foreground areas. For these reasons we preferred using a semi-automatic technique based on the SRM method.
The algorithm first automatically segments the image using a high merging factor, this leads to a very rough (over-merged) segmentation, but sufficiently precise for our aims. Second, an user-friendly GUI allows the user to easily select the foreground and the background regions.



a. The input image

b. The directional guidelines obtained by SRM

c. The foreground/background mask image

d. The output image (Tile size=5x10, snake size=2)

e. The foreground/background mask image

f. The output image (Tile size=5x5, snake size=0)

Figure 4. Some examples of opus vermiculatum mosaic.

This leads to a detailed foreground/background mask image which can be used in the successive steps. Note that this approach allows the user to produce several foreground/background mask images from the same input.

Then, the algorithm creates an "opus musivum" for the foreground regions and a regular grid of tiles for the background. Finally, it merges the two images in order to produce the "opus vermiculatum" of the input image. It is possible to perform an "a-priori" morphological operation of erosion on the mask image in order to produce the snake effect around the foreground features.
Figure 4 shows an example of the proposed technique; in particular Figures 4d and 4f show how the user can easily modify the final result by a different choice of the foreground/background mask image (Figures 4c and 4e, respectively) and of the snake size.

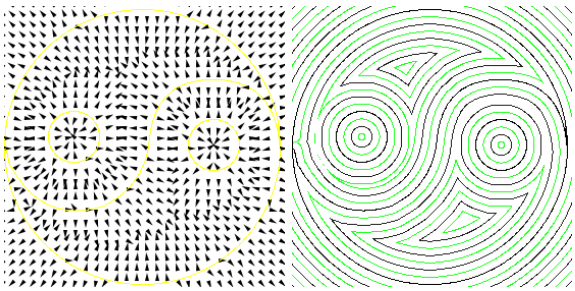
3.3 Foreground Rendering of Opus Musivum Mosaics

In this Subsection we briefly present the technique able to create "opus musivum" mosaics. Using the directional guidelines the algorithm first evaluates for each pixel of the image the distance transform [Har92], i.e. its minimum distance from any guideline pixel, obtaining a matrix (*dtM*). The use of the distance transform in the field of NPR was previously proposed by Gooch et al [Goo02], for a different purpose.
Starting from the distance transform matrix it obtains another two matrices: the gradient matrix (*gM*) and the level line matrix (*llM*):

$$gM(x,y)=\arctan\frac{dtM(x,y+1)-dtM(x,y-1)}{dtM(x+1,y)-dtM(x-1,y)} \quad (1)$$

$$llM(x,y)=\begin{cases}1 & if\ module(dtM(x,y),2\cdot tSize)=0\\ 2 & if\ module(dtM(x,y),2\cdot tSize)=tSize\\ 0 & elsewhere\end{cases} \quad (2)$$

where *tSize* is an user-selected integer value that denotes the size for the tiles. Their meaning is clearly shown in Figure 5 (in Figure 5b, black pixels have value 1, green pixels have value 2).



a. The gradient matrix

b. The level line matrix

Figure 5. Visual representation of the matrices used by the algorithm.

The algorithm is ready to place the tiles using the pixels in *llM* with value 2. Observe that such pixels form chain-like sequences. More precisely the algorithm proceeds as follows:

- while there are chains of pixels with value 2 not

yet processed:

a. select a chain;
b. starting from an arbitrary pixel on it "follow" the chain;
c. place new tiles at regular distances along the path (the orientation of the tiles is assigned using the gradient information from matrix *gM)*.

The distance along the chain that separates successive tiles is equal to *sSize* when tiles of dimension *tSize*x*sSize* have been adopted.

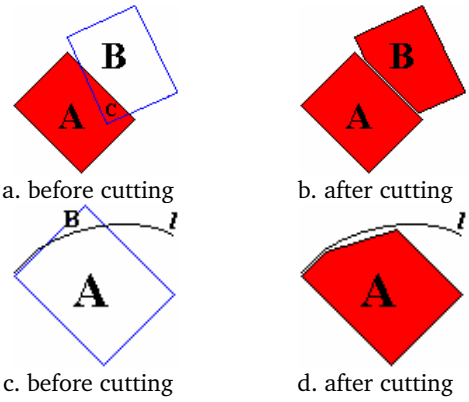
If tiles of fixed size and shape are positioned only according to the method described insofar two main difficulties arise:

1. tiles may overlap;
2. a single tile may cover an area across the "black pixels lines" (i.e. the pixels with value 1 in *llM*).

Both of these effects are unpleasant. In particular the problem in 2., completely destroys the guideline patterns and would result into blurred images.

To address these difficulties the algorithm adopts the following heuristic strategy:

1. the overlapping of tiles is easily detected maintaining a boolean mask of covered pixels. If a tile to be placed contains pixels already covered by previously placed tiles we change the original rectangular shape of the tile "cutting away" the overlapping pixels (see Figures 6a and 6b);
2. if a new tile crosses any "black pixel line" it is trimmed against this line (see Figures 6c and 6d).



a. before cutting b. after cutting

c. before cutting d. after cutting

Figure 6. Tiles cutting methodology.

Once the tile positioning and cutting phase has been completely carried out a couple of post-processing steps have to be performed in order to achieve a pleasant aesthetic effect.

First, "grout spaces" between tiles are important. To achieve the effect of cement showing through tiles a downscaling of each tile is done. This frees some pixels that will be assigned a unique color for concrete under the mosaic.

Second, the algorithm chooses for each tile to have a uniform color equal to the color of the pixel corresponding to its center in the source image. Other choices may lead to different artistic effects (for example a random perturbation of the selected color).

Major details can be found in [Dib05a].

4. EXPERIMENTAL RESULTS

To illustrate the effectiveness of the proposed technique we report both pictorial examples and some quantitative results. The algorithm has been implemented in Java2 Standard Edition 1.4.2 and all experiments have been carried out on a PC Athlon XP-M 1800+, 192MB RAM, with Windows XP Home Edition. To allow the real testing of the performances of the proposed technique an applet is available [Bat06a], at the same URL is also possible to download a JGimp plug-in and a Java application. Our implementation at this time does not rely on any particular graphic hardware acceleration.

Different parameters choices can lead to different final results. In our experiments, we used a merging factor for SRM equals to 3, a kernel size for the erosion morphological operation proportional to the number of lines of tiles snaking around the foreground and a Laplacian 3x3 kernel filter for the directional guidelines detection.

A mosaic made up of *N* tiles conveys more information than an image made up of *N* rectangular pixels in a regular lattice. Examples are shown in Figure 7 and 8. The proposed technique takes as input a digital image without any limitations on colors and resolution size. We found very appealing the results obtained using modern art sources like Cézanne's painting and Antonello da Messina artworks (Figures, 7c, 7d and 8, respectively). Figures 7a and 7b prove the effectiveness of the proposed technique when the source image is a good quality photograph.

Timing results (Table 1) show how the algorithm is fast enough to be used as a plug-in in a typical user-end software. It is straightforward to verify that the time asymptotic complexity is linear in the number of pixels of the image.

Size	Mosaic Mean Time (sec.)
600x600	10.485
800x600	12.689
593x886	15.182
1024x768	22.142

Table 1. Timing results.

5. CONCLUSIONS AND FUTURE WORKS

In this paper we presented a new method to create artificial mosaics achieving an impressive visual impact. Experimental results show the soundness of our algorithm.

There are several ways to improve the aesthetic of our results and several ideas started from this work:

1. a different strategy for choosing, in case of tile overlapping, which tile has to be cut. Heuristic rules or, perhaps, randomized choices could produce different outcomes;
2. automatic optimized choices of tile scale relative

to each input image is an open problem worth of further investigations;

3. generalization of our “mosaicists' heuristic” to other kind of primitive based on NPR processing seems possible and quite promising;
4. some generalizations as proposed in [Elb03], such as variable size tiles and photomosaic, are also considered for future work and research; it is also interesting to explore the possibilities offered using different basic shapes than rectangular tiles;
5. a different method to better find the directional guidelines is an important research investigation issue (see for example extensions proposed in [Noc05]);
6. exploitation of hardware graphics primitives to accelerate the mosaic synthesis;
7. extension of our method for mosaic rendering of 3D surface is probably the most exciting direction of research.

6. REFERENCES

[Ado06] Adobe Photoshop. http://www.adobe.com, 2006

[Bat03] S. Battiato, A. Pulvirenti, D. Reforgiato. Antipole Clustering for Fast Texture Synthesis. In proceedings of ACM/WSCG2003, 2003

[Bat06a] Battiato S., Di Blasi G., Farinella G.M., Gallo G. The Artificial Mosaic Creator applet www.dmi.unict.it/~gdiblasi/mosaic/mosaic.html, JGimp plug-in and Java application www.dmi.unict.it/~gdiblasi/mosaic/mosaic.jar, 2006

[Bat06b] Battiato S., Di Blasi G., Farinella G.M., Gallo G., Guarnera G.C. Ad-Hoc Segmentation Pipeline for Microarray Image Analysis. In proceedings of IS&T/SPIE Electronic Imaging 2006, 2006

[Can05] Cantone D., Ferro A., Pulvirenti A., Reforgiato Recupero D., Shasha D. Antipole Tree indexing to support range search and K-nearest neighbor search in metric spaces. IEEE Transactions on Knowledge and Data Engineering 17 (4), pp. 535-550, 2005

[Dib03] Di Blasi G., Reforgiato Recupero D. Fast Colorization of Gray Images. In proceedings of Eurographics Italian Chapter 2003, 2003

[Dib05a] Di Blasi G., Gallo G. Artificial Mosaic. The Visual Computer 21 (6), pp. 373-383, 2005

[Dib05b] Di Blasi G., Petralia M. Fast Photomosaic. In poster proceedings of ACM/WSCG2005, 2005

[Dib05c] Di Blasi G., Gallo G., Petralia M. Puzzle Image Mosaic. In proceedings of IASTED/VIIP2005, 2005

[Dob02] Dobashi J., Haga T., Johan H., Nishita T. A Method for Creating Mosaic Images Using Voronoi Diagrams. In proceedings of Eurographics2002, pp. 341-348, 2002

[Elb03] Elber E., Wolberg G. Rendering Traditional Mosaics. The Visual Computer, 19 (1), pp. 67-78, 2003

[Goo02] Gooch B., Coombe G., Shirley P. Artistic Vision: Painterly Rendering using Computer Vision Techniques. In proceedings of NPAR, pp. 83-90, 2002

[Hae90] Haeberli P. Paint by Numbers. In proceedings of SIGGRAPH1990, pp. 207-214, 1990

[Har92] Haralick R., Shapiro L. Computer and Robot Vision - Vol. 1. Addison-Wesley Publishing Company, 1992

[Hau01] Hausner A. Simulating Decorative Mosaics. In proceedings of SIGGRAPH2001, pp. 573-580, 2001

[Kap00] Kaplan C., Salesin D.H. Escherization. In proceedings of SIGGRAPH2000, pp. 499-510, 2000

[Kim02] Kim J., Pellacini F. Jigsaw Image Mosaics. In proceedings of SIGGRAPH2002, pp. 657-664, 2002

[Kim05] Kim K., Chalidabhongse T.H., Harwood D., Davis L. Real-time foreground–background segmentation using codebook model. Real-Time Imaging 11(3), pp 172-185, 2005

[Kle02] Klein A.W., Grant T., Finkelstein A., Cohen M.F. Video Mosaics. In proceedings of NPAR2002, pp. 21-28, 2002

[Kof22] Koffka, K. Perception: and introduction to the Gestalt-theorie. Psychological Bulletin 19, pp. 531-585, 1922

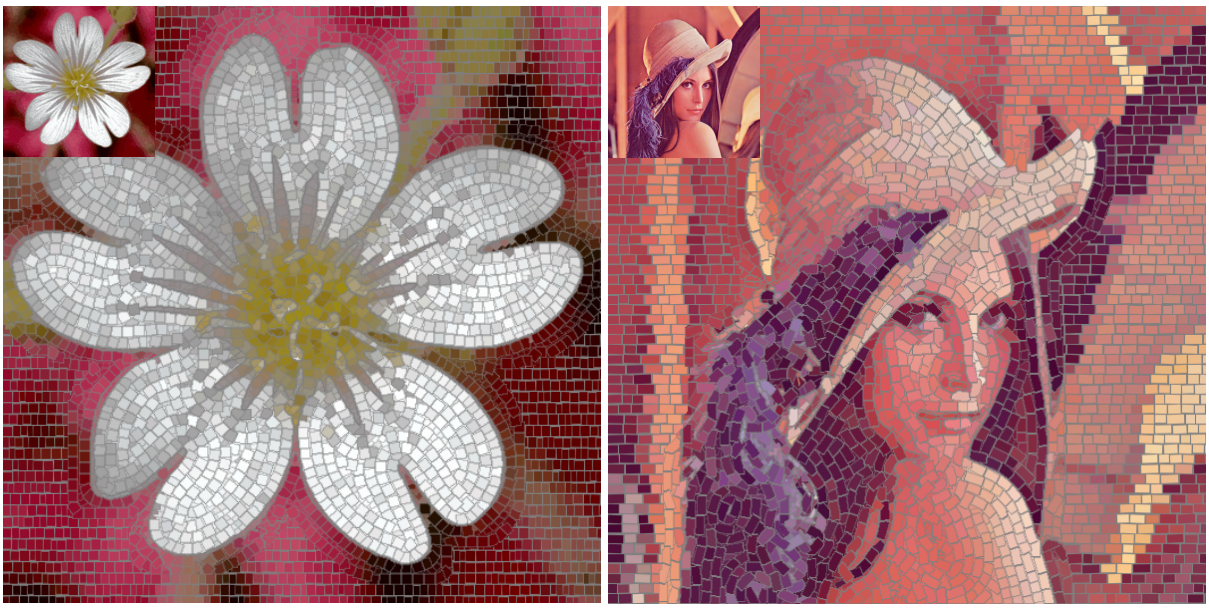
[Mee01] Meer P., Georgescu B. Edge Detection with Embedded Confidence. IEEE Transaction on Pattern Analysis and Machine Intelligence 23 (12), pp 1351-1365, 2001

[Noc04] Nock R., Nielsen F. Statistical Region Merging. IEEE Transaction On Pattern Analysis and Machine Intelligence 26 (11), pp. 1452-1458, 2004

[Noc05] Nock R., Nielsen F. Semi-supervised statistical region refinement for color image segmentation. Pattern Recognition 38 (6), pp. 835-846, 2005

[Pic04] Piccardi M., Jan T. Mean-shift background image modelling. In proceedings of ICIP2004, pp. 3399- 3402, 2004

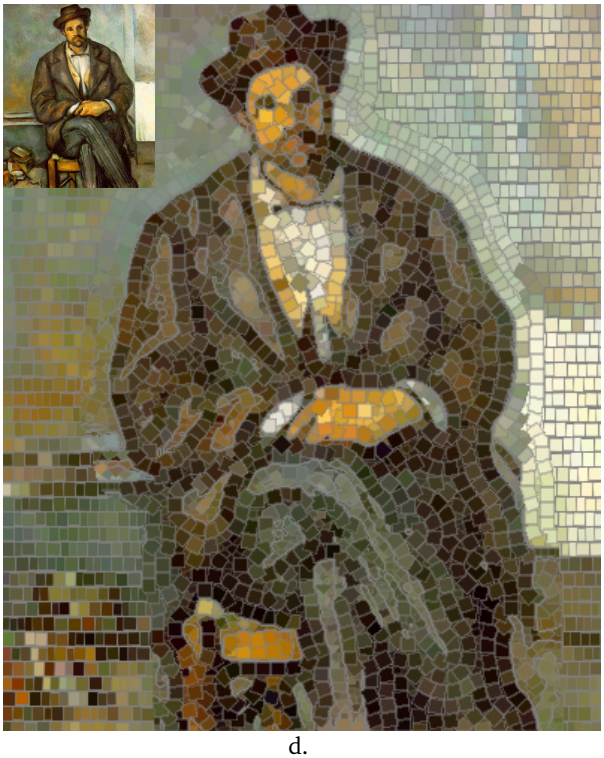
[Sil97] Silvers R., Hawley M. Photomosaics. Henry Holt, New York, 1997



a.

b.

c.



d.

Figure 7. Some examples of our algorithm applied on two typical test images (*a.* and *b.*) and two real well known Cézanne paintings (*c.* and *d.*).

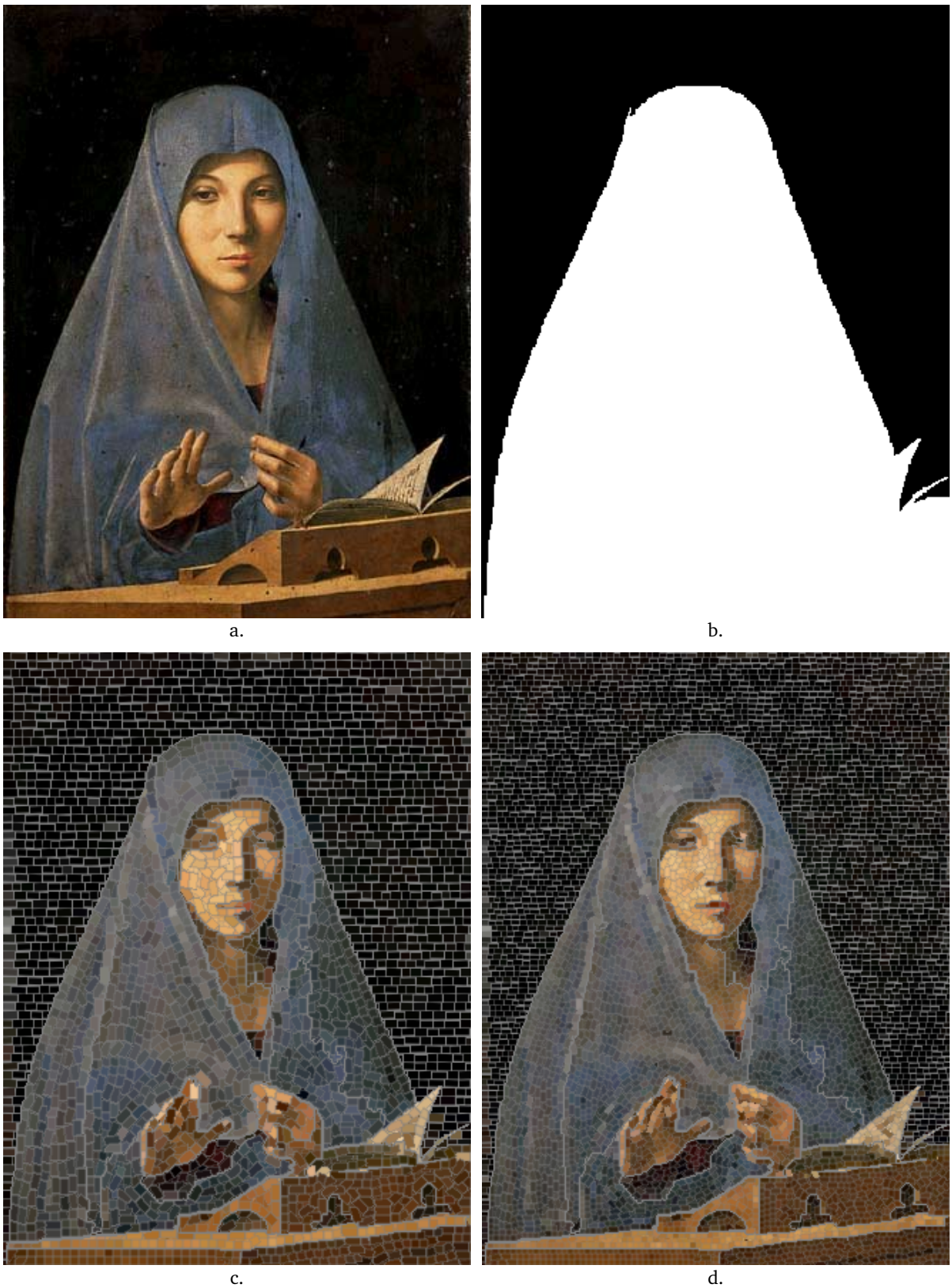


Figure 8. The Antonello da Messina painting *a.*, the background/foreground mask *b.* together with two different mosaics obtained using respectively the tile size 5x10 on *c.* and 3x6 on *d.*.