Interactive design of authentic looking mosaics using Voronoi structures

Lars-Peter Fritzsche[1] Heino Hellwig[1] Stefan Hiller[2] Oliver Deussen[2*]
[1] Dresden University of Technology, Germany
[2] University of Konstanz, Germany

Abstract

In this work we introduce new and efficient techniques to interactively create visually pleasing synthetic mosaics. We extend the known Lloyd's method for CVT computation to sets of small mosaic tiles. We use this improved placement algorithm in an interactive tool that enables the user to arrange tiles of various shapes and sizes, so that the set has the appearance of a traditionally handcrafted mosaic. The user controls the distribution process by adding contour lines and directional information. Tiles can be sized or shaped in order to better approximate the master image features. This interactive procedure allows for variations and visually more pleasing results. Additionally, it is less time expensive than using heuristic controlled automatic methods.

Keywords: mosaic illustrations, interaktive design, non-photorealistic rendering, Lloyd's method, CVT, NPR, computer graphic

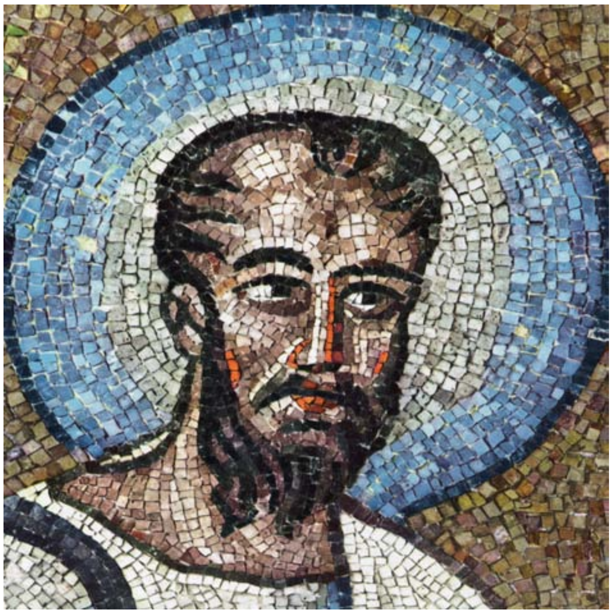


Figure 1. Traditional mosaic consisting of only six fundamental colors.

1. Introduction

The goal of computer graphics has been and still is the rendering of realistic appearing images. However in the area of non-photorealistic rendering, common drawing styles or illustration techniques, such as cross-hatching, are imitated. Also, traditional mosaics generate images with non-photorealistic characteristics. With our new method, we aim at simulating such traditional mosaic appearance. Realizing the importance of preserving features of classical art works, an interactive mosaic editor was implemented that enables the user to control mosaic placement in various ways. Mosaics have a history of more than 5000 years and were realized in many different styles and materials. The aesthetic pleasure of a mosaic results from the reduction of the visual data and the manual arrangement of important features. Some of the fundamental features are contour lines, colors, shapes, and positions of the basic primitives. These primitives are physical distinct sets of colored tiles that consist of either stone, ceramic or wood.

From the technical point of view, the reduction by fine sets of primitives is a bandwidth limited signal approximation, e.g. some kind of sampling. There are mainly two kinds of classical mosaic arrangement. Firstly, the abstract ornamental designs, and secondly the often complex portraits or illustrations of animals. In this paper we focus more on the latter and less on the mathematical motivated portraits. By studying the real world example at Figure 1 we can summarize a number

*LarsFritzsche@gmx.de,
Heino.Hellwig@inf.tu-dresden.de, hiller@uni-konstanz.de, deussen@uni-konstanz.de

of characteristic features:

- constant splices between the tiles;
- fundamental colors with stochastic variation in tone and luminance values;
- slight variation of tiles size and shape;
- tiles are arranged along feature lines of the underlying master image;
- smooth changes in tile orientation.

These features are necessary to keep inevitable visual artifacts at a low level, and are precisely the features we are aiming to effectively simulate with our proposed interactive editor.

Hausner [6] presented a technique to generate synthetic mosaics in an automatic way. It is based on modification of the known Lloyd's method to approximate centroidal Voronoi tessellation (CVT)[11] and distributes quadratic tiles in a mosaic like arrangement. Compared to classical mosaics (see Figure 1) the resulting pictures offer some flaws:

- some tiles appear misaligned;
- the splices are uneven;
- small tiles and large tiles are arranged together;
- the influence of the contour lines is too strong (see Figure 2).

Hausner's approach does not avoid these flaws, because his arrangement algorithm run heuristic controlled. Additionally, his methods reflect oversimplification and approximation, which prevents the user's input and restricts high quality output. This prevents the user from influencing the tile placement which is very important in order to create decorative mosaics of high quality.

2. Related Work

Haeberli [5] was one of the first authors that created non-photorealistic drawings by simulating stroke brushes. The image was segmented and every segment was represented by a stroke. For one of his styles, he used cones to simulate brush strokes. Each cone was assigned the color of the pixel under the cone peak at the underlying source image. In an orthographic projection this resulted in a colored tessellation of the canvas. The corresponding regions represented approximations of the Voronoi-regions around the cone peaks. Hoff et al. [9] used the same technique to generate Voronoi-diagrams using graphics hardware. This technique is widely known as cell based Voronoi approximation [11]. Deussen et al. [1] introduced the Lloyds method and by this the CVT(ref. also [3]) for non-photorealistic rendering to distribute points during the interactive generation of stipple drawings. Hausner [6] also applied a modified Lloyd's method to generate decorative mosaics. Small square tiles were distributed while applying the method. However, this method does not really approximate a CVT of the underlying tiles. In [7], Hiller extents the 2D Lloyd's method for CVT approximation to generator sets of heterogeneous simple polygons. By including the mass moments of polygons and the related Voronoi-regions as additional parameters into the Lloyd's iteration process, his method also optimizes the orientation of the polygon set.

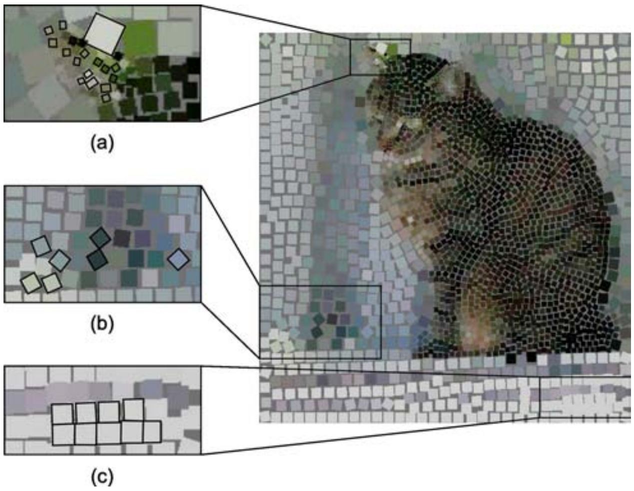


Figure 2. Synthetic mosaic generated by Hausner. Several artefacts can be seen such as wrong tiles sizes too close together (a), random misalignment of tiles (b), unwanted tile overlapping (c).

Several commercial tools offer so called mosaic-filters that allow the user to generate mosaics from images. The resulting pictures cannot be compared to traditional mosaics, because important image features are missing. Dobashi et al. [2] improve this kind of automatic mosaic generation by partitioning the image using Voronoi-diagrams.

They minimize the color differences between pixels in the reference and the Voronoi-regions. Here, the primitives are solely Voronoi-regions, and the resulting pictures appear more like stained-glass-style images than actual traditional mosaics, see Figure 3.

(a) (b)

Figure 3. Synthetic generated mosaics (a) by Dobashi et al. [2], (b) by Elber and Wolberg [4].

Observing that in classical mosaics the tiles are placed exactly along important feature curves, Elber and Wolberg [4] presented a technique for calculating offset curves to these feature curves. Adjacent to the offset curves, the rectangular tiles are placed tangentially to the curve in a very tight packing. A disadvantage of this method is that the resulting regular tile arrangement appears quite artificial, especially since the influence of the features and offset curves is not restricted, see Figure 3. In our approach we are able to model the influence of feature lines using our interactive tool.

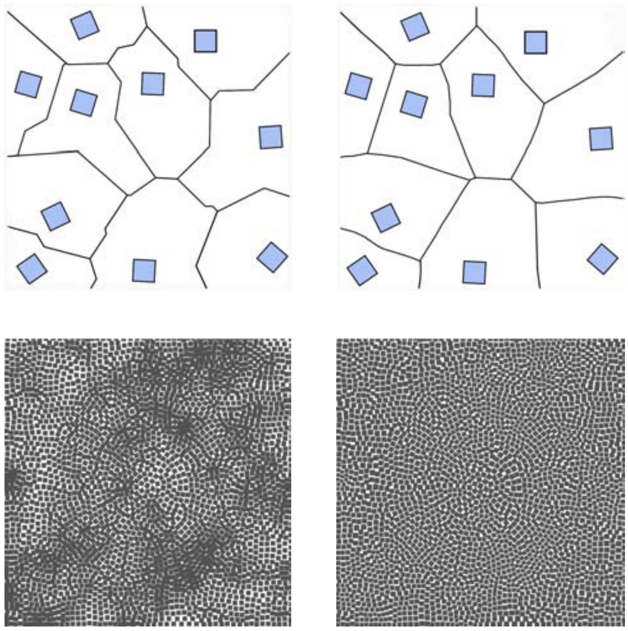
3. Mosaic generation

For an optimal placement of the mosaic tiles, we first discuss the existing methods and then give some suggestions for an improved version. Lloyds relaxation scheme for evenly distributing points can be noted as follows:

Input: set of (randomly) distributed points S
Output: set of points that generates a CVT
repeat
  1. compute Voronoi-Diagramm $V(S)$ of the point set
  2. move all points S_i to the center of gravity of their Voronoi region $V(S_i)$
until point movement is below a threshold ϵ or certain number of iterations steps reached.

In the above algorithm, a CVT is formed once all of the objects are placed in the center of gravity of their Voronoi-regions in such a way that the iteration converges.

As mentioned above, for a point set, the Voronoi-diagram can be computed using graphics hardware by placing cones at the position of the points in z-direction and analysing the resulting image from above (see [10, 9]). For our application the resulting approximation error is quasi negligible. Hausner uses a slightly modified version of this algorithm to distribute small square tiles. He extends the Voronoi-diagram calculation process by using a modified metric, which is in fact not really a metric, and definitively not a manhattan metric. Hausner uses a predefined directional field for tile alignment. The main problem of Hausner's method is that the tiles cannot be automatically placed in a artistic mosaic-like manner.



(a) (b)

Figure 4. a) Approximative Voronoi-diagram obtained by Hausner's method and (b) the exact solution using Hiller's method [7].

The proper arrangement of individual tiles is a highly artistic process. This is why we decided to model our solution as an interactive tool rather than simulating "‘art"’ with a set of heuristic methods. We found that heuristic driven methods very often produce unwanted artifacts such as miss-aligned tiles or highlighting of unwanted image parts.

3.1. Generalized algorithm

Hausner's method performs only a rough approximation, which in practice leads to overlaping of the tiles, if a tight packing is created (see Figure 4). If instead a frustum of pyramid with round corners is used, the shape of the Voronoi-regions can be much better approximated. For lines and polygons the correct depth buffer encodings for obtaining valid Voronoi tessellations are given in Figure 5. An improved version of Lloyd's method

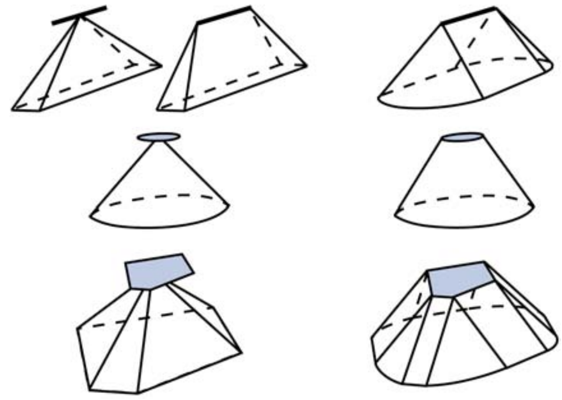


Figure 5. Representation geometries of line, circle, and polygon, left row Hausner's representation, on the right row our approximation.

that works with several kinds of objects was presented by Hiller et al. [7, 8].

For the actual work, we use a modification of this algorithm to distribute sets of heterogeneous mosaic tiles in a very tight packing evenly on the plane. In addition to previous work we control the tile packing (alignment, and the local tile density) by interactive defined feature-lines and polygons. Additionally we integrate an automatic method to control individually the tile sizes during the arrangement process, see following section.

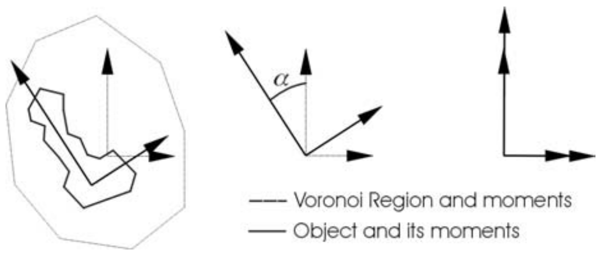


Figure 6. Rotation technique used in our extended Lloyds method by performing principle components analysis.

The generalized Lloyds Method for sets of polygons works as follows:

Input: Set M of random distributed and random orientated polygons P
Ouput: Set of polygons P', evenly distributed and orientated in context to each other
repeat
1. for each P approximate it's Voronoi-region by using geometric modelled distance
2. perform principle components analysis on polygons and their Voronoi-regions
3. compute center of gravity of polygons and their Voronoi-regions
4. move polygons until polygon CG and CG of Voronoi-region match
5. rotate polygons to bring principle-components together with that of the Voronoi-regions

until movement/rotation is below threshold ϵ.

In Figure 7 we show the results of this method in a simple example. We added components to this algorithm for better controlling its interactive behavior. We also integrated the directional information given by the feature lines to this new method.

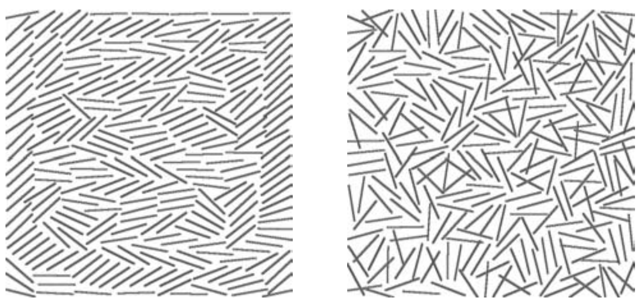
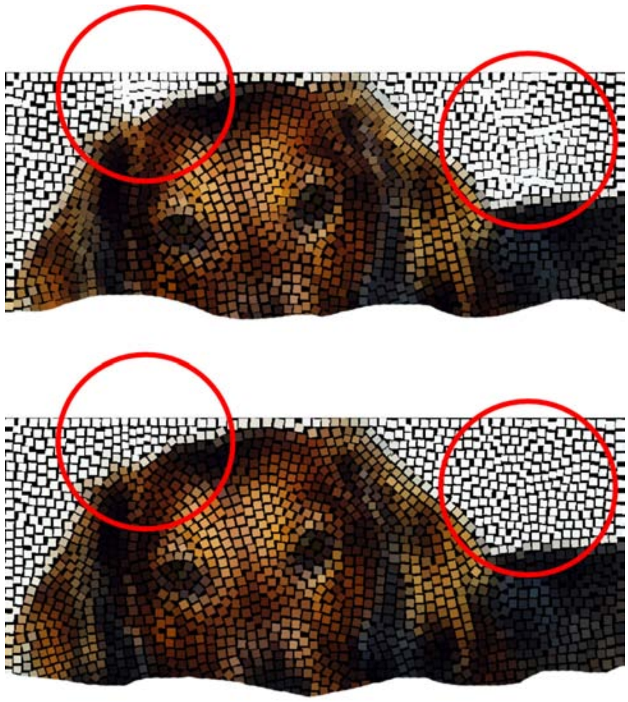


Figure 7. Relaxing a set of lines, on the left using our extended Lloyds method with rotation and on the right without.

3.2. Individual tile sizes

Also for our improved mosaic generation method, compactly arranged tiles tend to touch each other if the orientation of the tiles differs too much. We found that the general Lloyd's based relaxation method produces local varying densities. If the tiles are close together, like in our case, the results are clearly visible artefact. To avoid the corresponding visual artifacts, tiles are slightly changed



(a) (b)

Figure 8. Adjusting tile sizes: above original distribution with global correction; below individually corrected tile sizes.

in their size instead to perform additional time consuming relaxation steps. Hausner uses for this purpose a global formula for the tile size that affects all the tiles

$$d = \delta\sqrt{hw/n}$$

where hw is the total number of pixels, n the number of tiles, and δ a correction factor. A better solution is to additionally use an individual correction factor on the basis of the Voronoi-region belonging to the tile.

In our editor this is performed by measuring the size of the individual Voronoi-region. If $\bar{A}$ is the average size of all tiles, and $A_i = V(S_i)$, the individual size, we scale the size of the tile by

$$f = A/\bar{A}$$

In Figure 8 the effect is shown. While in 8(a) visual artifacts occur, the distribution of sub figure (b) looks better due to the small corrections of the individual tile sizes.

Additionally, in our editor the size of the tiles can be further reduced to represent important parts of an image by a larger number of now smaller tiles. This is performed by an additional image, which is imported into the editor. The grey scale values of the image control the size of the tiles at a local scale. We generate these control images in a fast and easy way by using different filtering methods included in bitmap editing tools like Gimp. Results are found in Section 5.

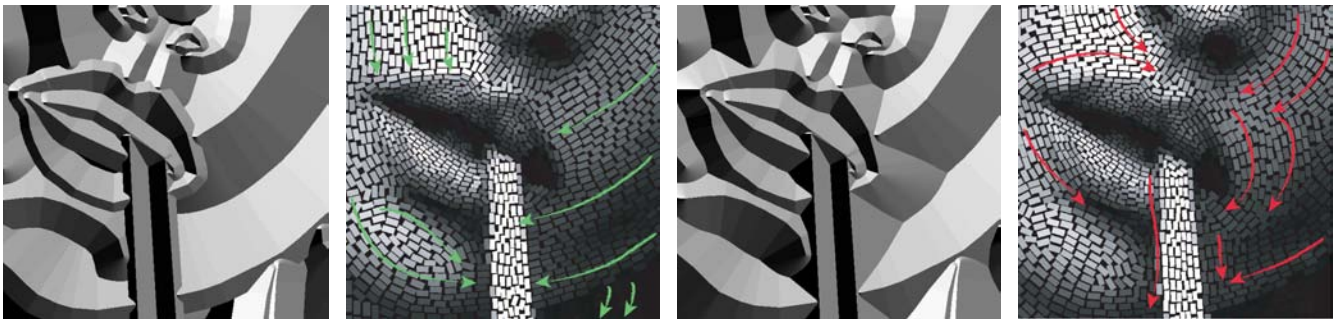
4. Interactive mosaic generation

As mentioned above, the real beauty of decorative mosaics can only be obtained by bringing about determining features of a given master image and incorporating them into the generation process. Our editor uses source images that can either be photo-realistic or sketch like.

The general procedure for the generation of mosaics using our method and tools, is as follows: to start, the user chooses a master image (a bitmap file), which shows the desired motif. The color values of this bitmap control the basic mosaic tile color, whereby a color adjustment of each individual tile is possible. Now the user defines feature lines and control polygons on the basis of this master image. To facilitate this, we have integrated the necessary interaktive functionality into the editor, e.g. basic vector drawing, editing and input/output functionality.

In direct context to the chosen motif, the artist highlights special parts of the picture. During this process the user defines the features lines by drawing with the mouse cursor over the features (edges) of the master image. Having completed this basic drawing, the area of influence on both sides of the feature lines are setup. The main purpose of the feature lines are to restrict mosaic-tile movement during the arrangement and relaxation, and to define a direction vector field for tile orientation.

Feature lines are modelled quite similar to the cell based Voronoi construction by 3D-geometry. This integrates the control information smoothly into the cell-based Voronoi based relaxation method. The realm of influence is for both sides of the lines (left and right) proportionally controllable. If all feature line are finally set, the direction vector field is constructed. This information is defined by the realm of influence of the feature lines. The basic direction is given by the orientation of the feature lines. However, it is also possible to define an offset angle to this preset direction. During the further design process the



(a) (b) (c) (d)

Figure 9. Controlling the direction of the tiles in the region around the feature lines: a) improved directions obtained by feature lines with varying influence regions; b) shows the resulting mosaic; c) automatic generation of feature lines; d) resulting mosaic for case c).

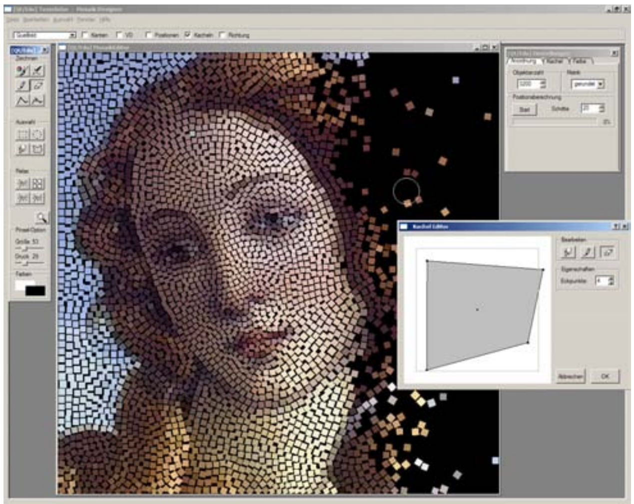


Figure 11. Editor while editing generated set of mosaic tile set, also the dialog for master tile shape editing is shown.

feature lines and their attributes are at all times changeable, the results can instantly be seen by the user.

Following the desired mosaic tile prototypes like circles, quads or user-defined n-edges are chosen. The user inputs the rough number of tiles to by inserted. First the insertion itself takes place automatically. After the basic tile set has been generated, tiles then can be deleted or inserted locally in an interactive way by using a brush like tool. Individual tile sizes are automatically adjusted to ensure equal gaps between the individual mosaic-tiles; this procedure is controlled by a globally set correction factor, see Figure 9.

The orientation of the mosaic tiles themselves, is iteratively executed with the extended relaxation technique, see previous section and [7]. In this procedure, the degree of uniform distribution is relative to the number of performed iterations.

For defining the direction field and the influence of our feature lines, we again use geometric representations, which can easily be embedded into our graphical Voronoi computation. As the user draws a feature line onto the image, the 3D representation geometry of the related influence function is defined. Thereafter the user is able to control the size of the geometry on both sides of the line. Figure 9 shows this effect. In Figure 9(a) the automatic placement of the tiles is performed. However, the directions of the tiles appear quite unnatural. In Figure 9(b) an improved result is obtained by using the influence regions.

After introducing feature lines and their influence regions, an additional Level-Of-Detail image can be used. This image controls the size of the tiles and therefore allows us to represent important parts of the image with a higher tile resolution. Usually, this image is a blurred and a color corrected version of the master image. It is obtained by image processing.

Now the user determines the number and shapes of the tiles. The tiles are placed and processed due to the presented extended version of Lloyds method. After determining the final positions, the color of the tiles is obtained from the source image, additionally a material texture can be applied to the tiles to achieve a more realistic appearance.

5. Examples

In Figure 10 we demonstrate the difference between Hausner's placing scheme, and various results of our new method in which we used refined tile orientations and sizes. Using this interactive editor, the user is able to artistically interpret the master image precisely according to his personal requirements. Additionally, the user controls the clarity and details of the mosaic representation. If material textures are added, a good approximation of a real world mosaic is achieved.

Figure 12 shows two mosaics consisting of polygonal tiles. In both cases the improved version of Lloyds method was used to place the mosaic tiles aesthetically. Also the tiles sizes were corrected due to the proposed individual method. In Figure 12(b), additionally a texture was a applied.

In Figure 13 a complex mosaic is shown. It consists out of six thousand tiles which were arranged using our editor. Level-of-detail was used to represent the face by a larger number of small tiles, whereas the crown and the background was created using larger tiles.

6. Future work

Working with the mosaic editor, we are convinced that much power lies in the interactive generation. We will extend the tool set of the editor by adding new features that allow the user to gain the control over the tiles. The mosaic editor should also be used for prototype designs of real world mosaics. Also other non-photorealistic rendering techniques may be realized with parts of our methods like cross-hatching.

References

[1] O. Deussen, S. Hiller, K. van Overveld, and T. Strothotte. Floating points: a method for computing stipple drawings. *Computer Graphics Forum*, 19(4):40–51 (Proceedings of Eurographics), 2000.

[2] Y. Dobashi, T. Haga, H. Johan, and T. Nishita. A method for creating mosaic images using voronoi diagrams. *Proceedings of Eurographics 2002 (Short Presentations)*, pages 341–348, 2002.

[3] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations. *SIAM Review*, 41(4):637–676, 1999.

[4] G. Elber and G. Wolberg. Rendering traditional mosaics. *The Visual Computer*, 19:67–78, 2003.

[5] P. Haeberli. Paint by numbers: Abstract image representation. *SIGGRAPH Conf. Proc.*, pages 573–580, 1990.

[6] A. Hausner. Simulating decorative mosaics. In *SIGGRAPH Conf. Proc.*, pages 573–578, 2001.

[7] S. Hiller and O. Deussen. Voronoi-relaxierung allgemeiner objekte. In Stefan Schlechtweg Thomas Schulze and Volkmar Hinz, editors, *Proc. Simulation und Visualisierung Magdeburg*, pages 223–234. Erlangen cdot Ghent, SCS Europe, 2001.

[8] S. Hiller, H. Hellwig, and O. Deussen. Beyond stippling - methods for distributing objects on the plane. *Proceedings of Eurographics*, 2003.

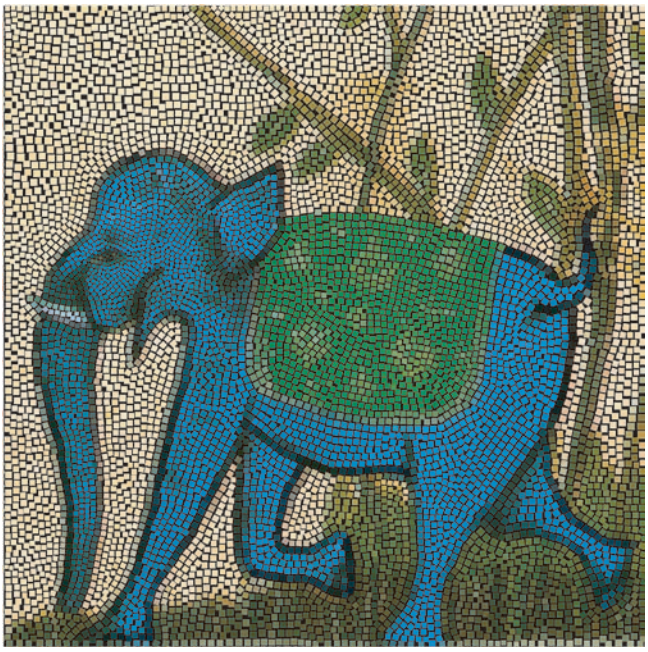
[9] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manoc ha. Fast computation of generalized voronoi diagrams using grap hics hardware. In *SIGGRAPH Conf. Proc.*, 1999.

[10] J. Neider, T. Davis, and M. Woo. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley, opengl version 1.2 edition, 1999.

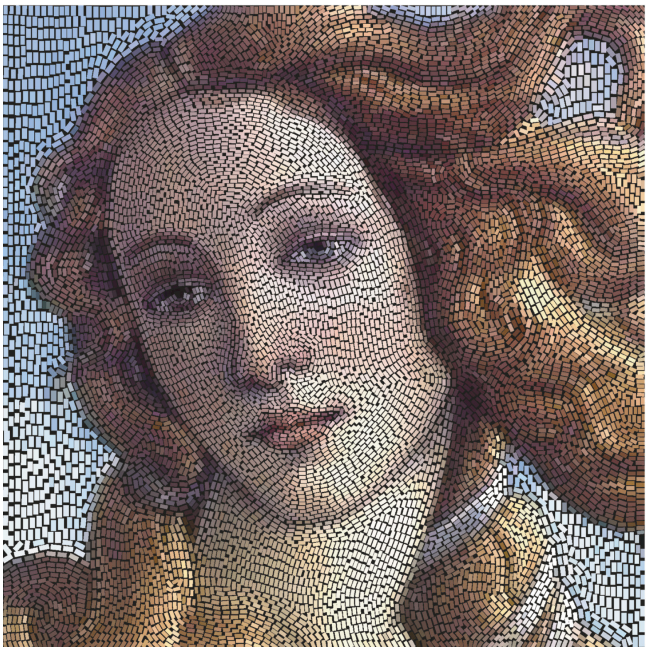
[11] A. Okabe, B. Boots, K. Sugihara, and Sung Nok Chiu. *Spatial Tessellations Concepts and Applications of Voronoi Diagrams.* John Wiley & Sons, 1999.

(a) (b) (c) (d)

Figure 10. Differences in mosaic generation algorithms: a) Hausners method; b) improved tile placement using the proposed method; c) feature lines and size variation; d) textured version.

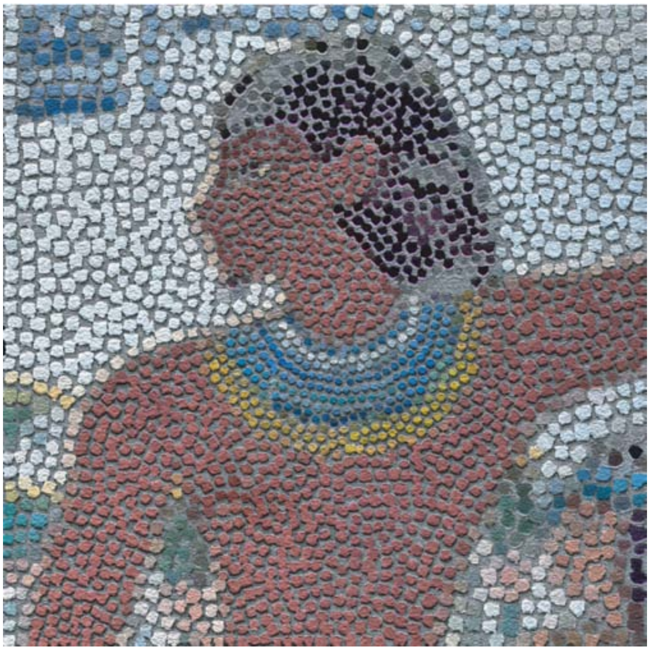


(a)



(b)

(c)



(d)

Figure 12. Two mosaics with arbitrary shapes. a) elephant; b) venus; c) version with level-of-detail; d) textured tiles increase realistic impression.

Figure 13. Justinian, source image obtained from an ancient mosaic which was converted into a smooth image and then re-tesselated.

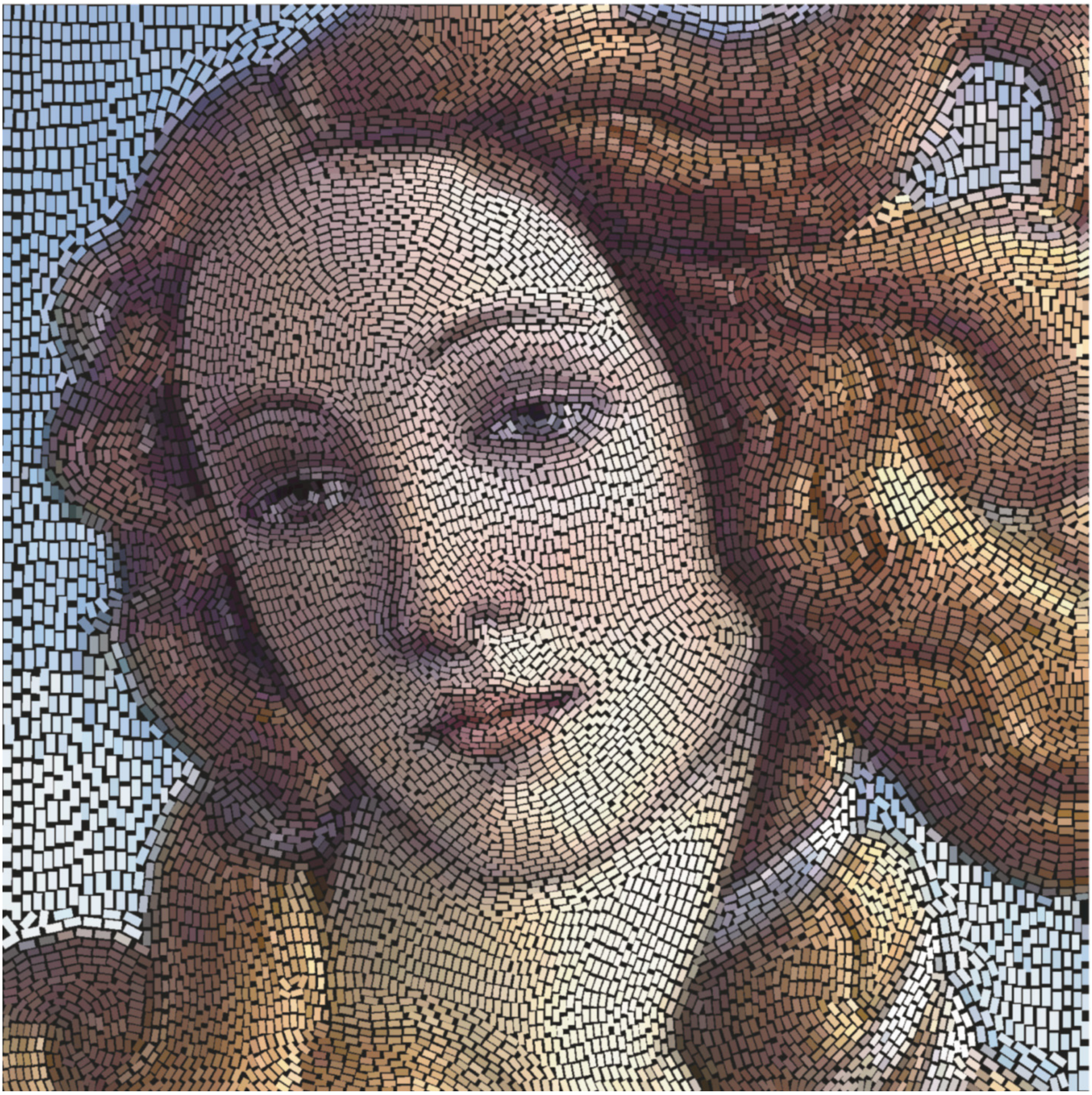


Figure 14. Venus, created from 10.000 rectangular tiles