Modeling the Copacabana Sidewalk Pavement

Tatiana Waintraub
Computer Science Department, PUC-Rio
Rio de Janeiro, Brazil
Email: tatianawaitraub@gmail.com

Waldemar Celes
Tecgraf, Computer Science Department, PUC-Rio
Rio de Janeiro, Brazil
Email: celes@tecgraf.puc-rio.br

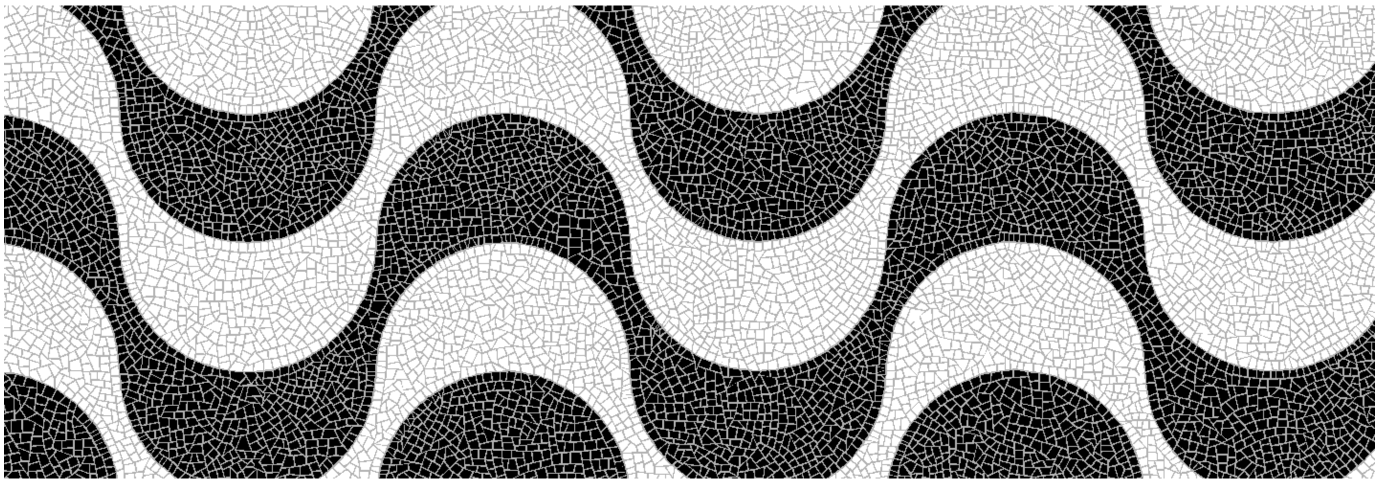


Fig. 1. An image of the Copacabana sidewalk modeled with the proposed technique

***Abstract*—In this paper, we propose a method to model the Copacabana beach sidewalk pavement, and Portuguese pavements in general. Given a black and white source image, the proposed method outputs the geometry of all individual stones that compose the pavement. Different from previous mosaic techniques, we focus on capturing the particularities of such pavements: the stones (tiles) must completely follow the edges, being mostly represented by irregular quadrilaterals with no particular orientation. A set of experiments demonstrates the effectiveness and stability of our proposal.**

***Keywords*-mosaic; procedural modeling; Voronoi diagram;**

I. Introduction

The Copacabana beach sidewalk, in Rio de Janeiro, represents a famous use of a traditional pavement style known as Portuguese pavement. This paving technique is used for many pedestrian areas in Portugal and in the old Portuguese colonies, including Brazil. The pavement is created by setting black and white stones, in general, in an harmonic way revealing distinctive tiled patterns. The Copacabana beach sidewalk, in its current form, was designed by Roberto Burle Marx, a famous Brazilian landscape architect. The pattern exhibits large curves resembling the waves of the sea, as shown in Fig. 2.

Our purpose is to devise an unsupervised computational method to model the Portuguese pavement as presented in the Copacabana sidewalk. Portuguese pavement is in fact a kind of Opus Palladium mosaic, where tiles of irregular shapes are used to convey images in an expressive manner. There are several proposals in the literature for computing digital mosaics using different strategies [1]. However, observing the Copacabana sidewalk pavement, one can note a few distinct features that have to be considered for our purpose:

- The stones present mostly irregular quadrilateral shapes
- The region bounds (edge features) are perfectly honored
- The stones (tiles) are placed with arbitrary orientations except near the bounds

This paper presents a method capable of modeling Portuguese pavement in an effective way. Inspired by techniques employed in previous works [2], [3], [4], [5], [6], such as hardware-assisted centroidal Voronoi diagram and distance field, our method does results in mosaics with the characteristics presented by the Copacabana beach sidewalk. Figure 1 illustrates the achieved result. This paper also demonstrates that the presented technique can be directly applied to model other occurrences of Portuguese pavement.

The rest of this paper is organized as follows. The next section reviews related works. Section III describes the proposed method in detail. Section IV presents and discusses achieved results, and concluding remarks are drawn in Section V.

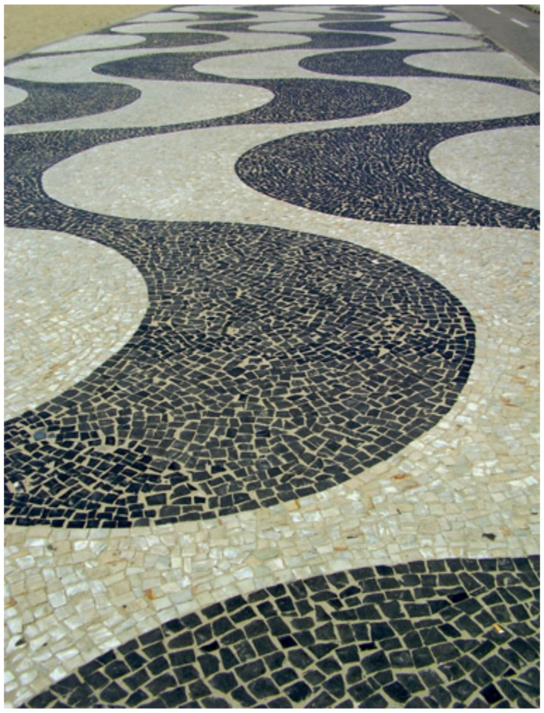


Fig. 2. Photo of Copacabana beach sidewalk (image from Wikipedia.org)

II. Related work

Battiato et al. [1] presented a nice and comprehensive overview of different digital mosaic techniques. They have grouped mosaics in two types: *tile mosaics*, where a source image is decomposed into tiles, and *multi-picture mosaics*, where images from a database is used to cover an assigned source image. Clearly, the Copacabana sidewalk fits in the first group, using as the source image the black and white drawing of waves.

Different proposals use computational geometry combined with image processing to achieve the mosaic patterns. Haeberli [2] inspired many other proposals by using a Voronoi diagram from randomly placed generating points. Voronoi diagrams produce mosaics with tiles of variable shapes and does not honor region bounds; on the other hand, Voronoi diagram can be efficiently computed with the use of graphics processing units [7]. In order to honor region bounds, Dobashi et al.[8] integrated edge information to the Voronoi diagram construction. They proposed an iterative procedure that tries to minimize an error function by repositioning the center of the polygons produced by the Voronoi. Faustino and Figueiredo [9] used centroidal Voronoi diagram together with a density function to adapt the size of the tiles according to features of the source image; in their proposal, the tiles are not aligned to the image edges.

Centroidal Voronoi diagram tends to generate regular hexahedral grids. To avoid such a pattern, Hausner [3] proposed to build centroidal Voronoi diagram with the use of a Manhattan-like metrics. They built the diagrams with graphics hardware acceleration by drawing square based pyramid with apex at the seed points. Each pyramid is aligned according to a direction field computed from the source image. In the end, they replaced the Voronoi cells by regular square tiles; as a result, the tiles were laid along curving square grids. Elber and Wolberg [4] also use Manhattan distance metrics to lay rows of square tiles aligned to feature curves of the source image.

Di Blasi and Gallo [5] have also proposed an effective method to build artificial mosaics. From the source image, they first extracted the guidelines, filled the distance transform matrix, and then computed the corresponding gradient matrix, used to guide tile placement, leading to realistic results. Fritzsche et al. [6] have opted to implement an interactive tool in order to better achieve artistic mosaics, and employ centroidal Voronoi diagram with Lloyd's method. More recently, Zhang and Yu [10] have proposed a technique to create mosaics with irregular tiles by employing polygon tessellation.

Although expressive, none of these techniques can be directly employed to reproduce the Portuguese pavement as found in the Copacabana sidewalk. Different from the previous proposals, the source image in our case is rather simple; however, this simplicity cannot lead to unnatural stone placement.

Passos and Walter [11] proposed a technique to build mosaics on 3D surfaces with arbitrarily-shaped tiles, combining a physically-based relaxation method with Voronoi diagram. In order to honor region bounds, they introduced artificial forces to repel tiles from the edges. These artificial forces help to avoid placing tiles across region bounds but tiles do not perfectly follow the edges. In the end, they achieved 3D textured mosaics with variable-shaped tiles; however, the final arrangement is not suite for our purpose; as Voronoi cells are directly mapped to mosaic tiles, the final result tends to include several close to regular pentagonal and hexahedral tiles.

III. Proposed method

The proposed method to model Portuguese pavements is also based on the construction of a centroidal Voronoi diagram (CVD). We have also opted for using graphics hardware acceleration to efficiently compute the diagram. In order to ease the implementation, we decided to build the diagram in the same resolution of the provided source image. The number of stones, n, could be explicitly provided, but we prefer to derive n from the image resolution. By doing that, we ensure screen-coordinate precision for accurately extracting the geometry of the stones.

We derive n by first choosing the average amount of pixels that represent half of each stone side, $\bar{h}$. Assuming at first each stone as a regular square, we have:

$$n = \frac{I_w \, I_h}{(2\bar{h})^2} \tag{1}$$

where I_w and I_h represent the width and the height of the source image, respectively. As we shall describe, the average half side, $\bar{h}$, plays an important role as a parameter throughout the proposed method.

Once the number of stones is defined, the proposed method performs the following procedures to model the pavement:

- Compute the corresponding distance field
- Compute the centroidal Voronoi diagram
- Extract and adjust the stone shapes

A. *Distance field*

The input source image is a black and white image representing the mosaic drawing. One typical image is illustrated in Fig. 4a. The first step of our method is to compute a distance field from the source image. The distance field will express the distance from any point in the domain to the closest edge, i.e., the frontier between black and white pixels. The distance field is represented by an image of the same resolution of the source image.

We first initialize each pixel value of the distance field image as a huge number. We then process each corresponding source image pixel and check if it is on an edge; if so, we assign the distance value to the pixel. This is performed by checking the vicinity of each given pixel. Two groups of vicinities exist: four adjacent vertical or horizontal pixels and four adjacent diagonal pixels. If there exists a pixel with different color in the first vicinity group, the distance value of the current pixel is set to 0.5; otherwise, if there exists a pixel with different color in the second vicinity group, the distance value is set to $\sqrt{2}/2$. If there is no pixel with different color in the vicinity, no value is set. To fill the missing values, we apply a chamfer distance transform using the quasi-Euclidean 3x3 chamfer type [12]. Fig. 3 illustrates this process, and Fig. 4b shows the corresponding computed distance field image from the source image shown in Fig. 4a.

The distance field values will be used to evaluate the distance of each stone to the closest edge. We also need to compute the corresponding gradient of the distance value to align the stones near the edges for the Voronoi computation. The gradient value is evaluated using the Sobel operator:

$$g_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad g_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (2)$$

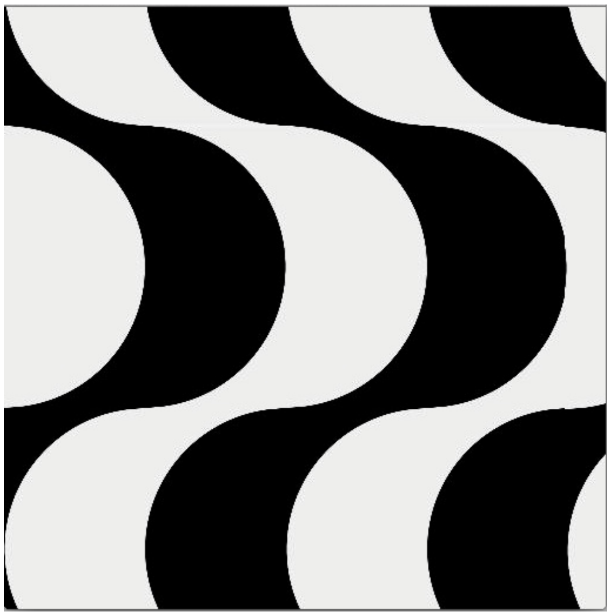
	0.707	0.500	0.500
0.707	0.500	0.500	0.500
0.500	0.500	0.707	
0.500	0.707		

(a) Initial values

1.207	0.707	0.500	0.500
0.707	0.500	0.500	0.500
0.500	0.500	0.707	**1.000**
0.500	0.707	**1.207**	**1.414**

(b) Final values

Fig. 3. Distance field computation using chamfer distance transform


(a) Source image

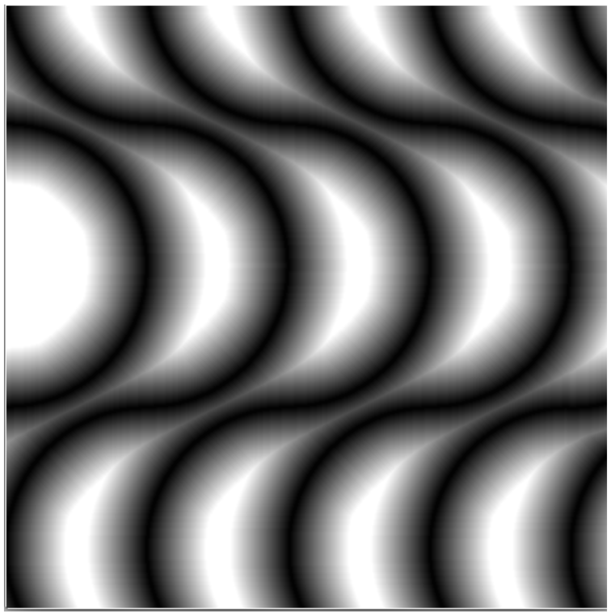

(b) Distance field

Fig. 4. Distance field extracted from the source image

B. *Centroidal Voronoi diagram*

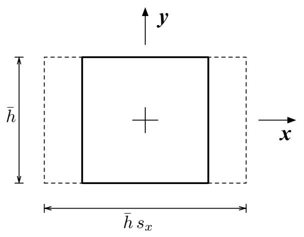
The next step in our algorithm is the computation of the centroidal Voronoi diagram (CVD). The final cells of the CVD are mapped to model the stone shapes. For that reason, we have the challenge to build the CVD in a way that the resulting cell shapes attend, as much as possible, the requirements imposed by the particularities of the Copacaba-sidewalk mosaic (and usually Portuguese-pavement mosaic in general): irregular quadrilateral shapes, honored bounds, and random orientation where possible.

To compute the CVD, we first randomly distribute the generating points in the domain and use the graphics hardware to compute the diagram. The CVD results from an iterative method. At each pass, for each resulting Voronoi cell, we compute its center of mass and use it as the new corresponding generating point. This method produces a stable diagram after a few iterations.

The conventional CVD tends to result in a regular hexahedral grid of cells. In order to avoid this arrangement, we employ the same strategy as Fritzsche et al. [6]: the use of frustums of pyramid with round corners, instead of cones, as primitives to compute the diagram on the graphics hardware. Each frustum of pyramid is assigned a color that encodes the generating point identification number. In our method, each frustum of pyramid is created based on a axis-aligned rectangle. For each generating point, a different rectangle is created, randomly choosing a scale factor, s_x, in the x direction as illustrated in Fig. 5.

When drawing the pyramid for the diagram construction, in order to get cells with no particular alignment direction, we apply a rotation along the z axis; the angle of rotation is chosen based on a 2D Perlin noise distribution. In this way, we tend to get cells with arbitrary orientation while preserving alignment among adjacent cells.

The main challenge to compute the CVD using this non-conventional metrics is to choose the initial rectangular size of each frustum cap. If a too small size is chosen, the tendency is that we end up with a regular CVD, because the cap is so small that could be replaced by a single point (i.e., the pyramid would be replaced by a cone). On the other hand, if a too large size is chosen, we may have overlapping caps in the first place, invalidating the diagram construction. We have opted to use the average half stone side (see Eq. 1) as a parameter: the cap size is set proportional to the value of $\bar{h}$, as shown in Fig. 5.



(a) Pyramid cap

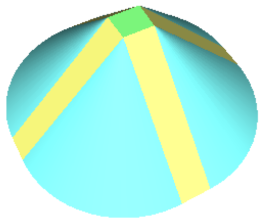
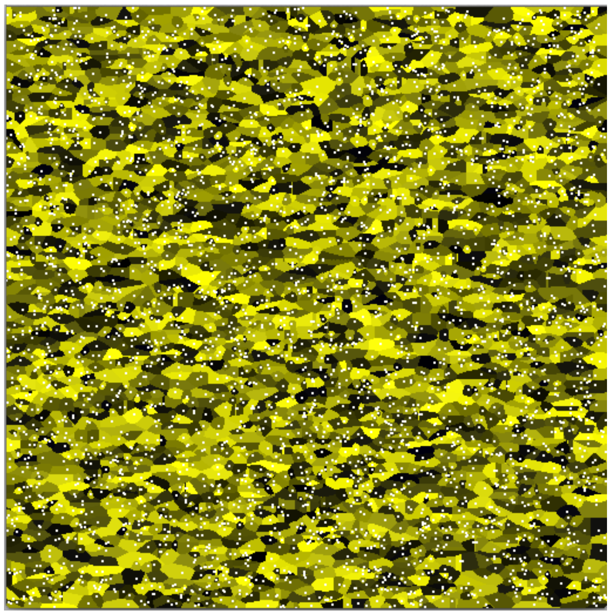

(b) 3D pyramid

Fig. 5. Pyramid primitive for Voronoi computation

The value of $\bar{h}$ is also used to make the cells of the Voronoi diagram to naturally follow the image edges. During the iterative procedure, after computing the center of mass of each cell, we fetch the distance field image and check if the point is too close to an edge. If the distance to the closest edge is less than $1.5\bar{h}$, we compute the corresponding gradient (Eq. 2) and moves the point along the gradient direction in order to set it at a distance equals to $\bar{h}$ to the closest edge. As a result, all cells along the edges will have their generating points aligned. To ensure that cells across edges meet each other on the edges, we do not use the value from the Perlin noise to rotate the corresponding pyramids, but instead we use the distance-field gradient to set a rotation angle that aligns the cells with the edges:

$$\theta = \texttt{atan2}\left(\frac{g_y}{g_x}\right) - \frac{\pi}{2} \tag{3}$$

Figure 6 illustrates the iterative construction of the CVD based on the source image shown in Fig. 4a. Note that, in the final configuration, the cells are aligned to the image edges,


(a) Initial configuration

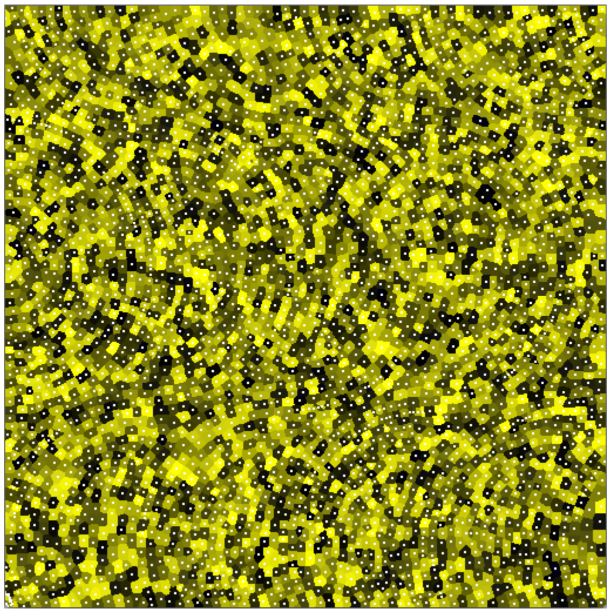

(b) Final configuration after 20 iterations

Fig. 6. Iterative centroidal Voronoi diagram construction

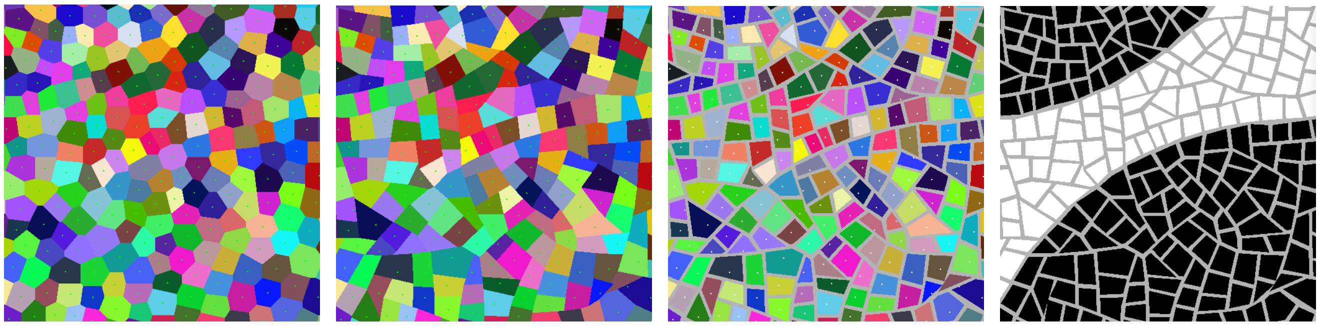

(a) Polygons from Voronoi (b) Wide angle elimination (c) Imposed separation (d) Final black and white stones

Fig. 7. Polygon extraction

present rectangular-like shapes, and are placed with random orientation except near the edges.

C. Polygon extraction

Once the Voronoi diagram is built, we start the procedure to extract the geometry of each stone, based on the diagram cells. In the CVD image, each cell is identified by pixels with a particular color, which encodes the cell identification number. Our goal is to model the set of stones, with the corresponding vertex coordinates.

We first process the CVD image to identify the vertices. Given a pixel $p_{(i,j)}$, we access its three adjacent pixels $p_{(i+1,j)}$, $p_{(i,j+1)}$, and $p_{(i+1,j+1)}$. Among these four pixels, we check how many different colors exist: if there exist 3 or 4 different colors, the middle point, $(x, y) = (i + 1, j + 1)$ in world coordinate, corresponds to a vertex. This vertex is saved and added to the vertex list of each polygon identified by the different colors. Note that vertices are shared among adjacent polygons. For pixels along the border of the image, a similar procedure is applied looking for 2 different colors among the two adjacent pixels. The four corners of the image also represent vertices assigned to the polygons corresponding to the colors of the corner pixels.

Next, based on its vertices, we compute the center of mass of each polygon and then sort the incidence to get a cyclic counter-clockwise sequence, based on the angle of the vector from the center of mass to each vertex (we assume cells are star shaped). A set of extracted polygons, corresponding to the diagram cells, is illustrated in Fig. 7a.

As can be noted, polygons from the Voronoi cells still do not attend our requisites. Our goal is to get a tessellation where most tiles are quadrilaterals. We then process the list of extracted polygon performing a set of procedures to converge the number of vertices per polygon to four, as far as possible, without corrupting the achieved tessellation.

The first employed procedure eliminates the wide angles of polygons. For each polygon, if we find a wide internal angle (for instance, greater than 120°), we move the vertex to the segment connecting the previous and the next vertices in the vertex list of the polygon. We in fact turn the angle to a value equals to 180°; the vertex cannot be eliminated from the tessellation due to the adjacent polygons. The image in Fig. 7b illustrates the change in polygon shapes.

Voronoi cells do not perfectly honor region bounds; moreover, moving the vertices can lead to edge misalignment. In order to ensure that the tessellation completely follows the edges, we move, along the distance field gradient towards the edge, all vertices whose corresponding source image color differs from the color associated to the polygon's center of mass. Note that these procedures move vertices that are shared by all adjacent polygons, avoiding overlapping.

We then consider each polygon in isolation, replicating the shared vertices, and extracted the final list of vertex coordinates. For each individual polygon, vertices associated to wide angles are no longer considered. We also eliminate very small edges, discarding one of its incident vertices. Finally, we shrink the polygon by moving each of its vertices towards the center of mass, by half of a provided grout separation value. Note that moving the vertices towards the center of mass does not shrink the polygons in an envelope way. This in fact is a desired behavior, since it mimics the grout thickness variation found on real pavements. The image in Fig. 7c illustrates the achieved results of this procedure.

It is also important to note that, in all these procedures, we do not allow the number of vertices of a polygon to be reduced to a value less than four. As a final step, as illustrated in Fig. 7d, we assign black or white color to each polygon (stone), based on the source image color associated to the center of mass.

IV. Results

This section presents some computational experiments that demonstrates the proposed method in action. All these experiments were run setting the parameters to the following values:

- the average half stone side, $\bar{h}$, is set to 4.0;
- the scale factor, s_x, applied to the pyramid cap, varies from 0.5 to 2.0;
- the number of iteration to compute the CVD is set to 20;
- the wide angle limit to eliminate the polygon vertex is set to 120°;

The image in Fig. 8 is the achieved result from the source image shown in Fig. 4a. Together with the image in Fig. 1, it demonstrates that the proposed method is able to model the Copacabana beach sidewalk with its famous waves. The stones follow the edges with no significant misalignment. The stones are placed with no particular orientation, except near the edges, mimicking the arrangements we find in real Portuguese pavements.

An important contribution of our method is its ability to create mosaics where most tiles have irregular quadrilateral shapes. This is achieved by first using frustums of pyramid with rectangular caps to build the CVD and Perlin noise to randomly set cell's orientation without degrading adjacent coherence. The resulting cells then undergo a set of procedures to model the final shapes of the stones. The histogram shown in Fig. 9 depicts the distribution of number of vertices per polygon: the initial distribution corresponds to the configuration just after converting CVD cells to polygons; the final distribution corresponds to the configuration of the black and white stones. As can be noted, most of the stones are quadrilaterals.

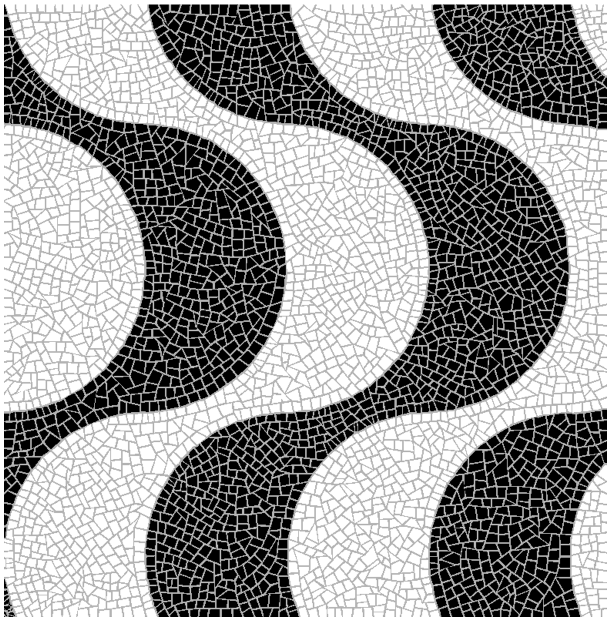


Fig. 8. Modeled Copacabana beach sidewalk

In Fig. 10, it is shown the result from modeling a real pavement drawing found in Lisbon. Again, one can note that the proposed method does correctly model the stones of the pavement.

The method can be applied to general drawings in order to get a Portuguese-pavement mosaic style. See the flower and the guitar drawings, and the image of the statue of Christ the Redeemer, modeled in Fig. 11. These results illustrate how generated mosaics honor the source image edges.

The fact that the same set of parameter values works for a variety of different experiments demonstrates that we have achieved one of our goals: the conception of an unsupervised method for modeling Portuguese pavements. This also demonstrates that the method is quite stable. Nevertheless, it

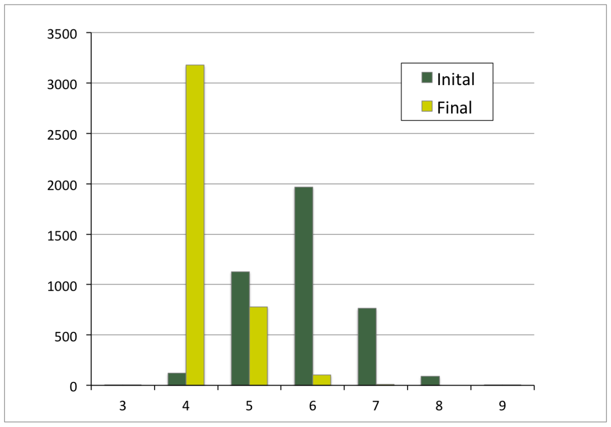


Fig. 9. Histogram of number of polygons: the proposed procedure produces tessellation where the majority of polygons are quadrilaterals

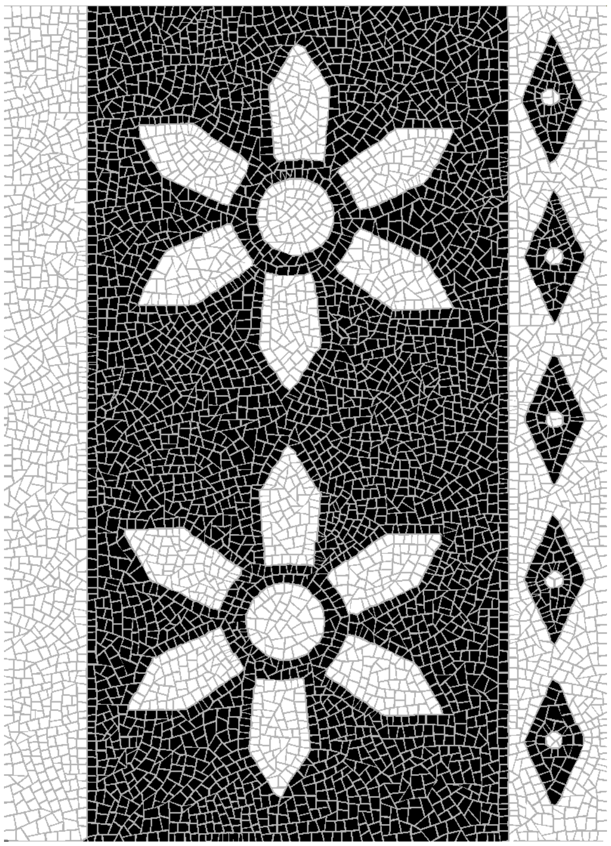
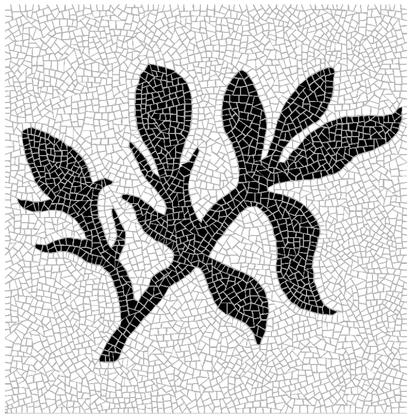
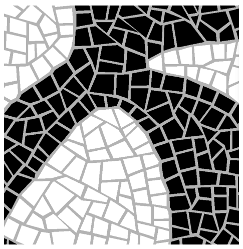
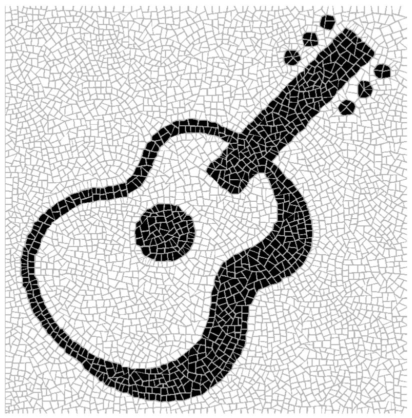


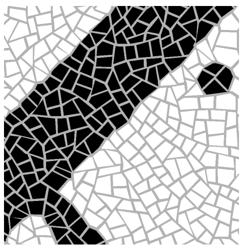
Fig. 10. Lisbon sidewalk

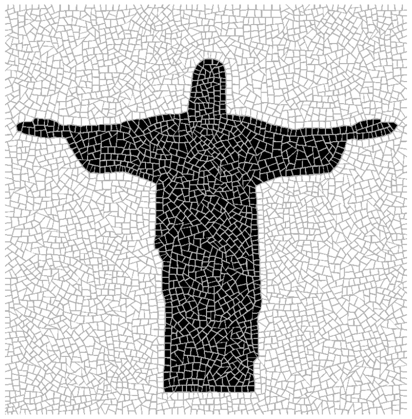
may be desired to adjust these parameters to get different arrangements; especially, the first two that control the sizes and influence the shapes, respectively, of the resulting stones. As an example, we rerun the algorithm using the source image in Fig. 4a but fixed the scale factor (s_x) to 1.0. In Fig. 12, we compare the histogram of stone areas. As can be noted, a constant scale factor produces more stones with similar size (area), as expected.


(a) Flower drawing


(b) Flower zoom


(c) Guitar drawing


(d) Guitar zoom


(e) Statue of Christ the Redeemer

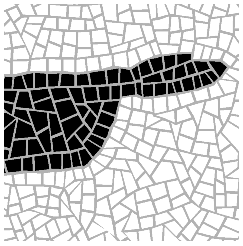

(f) Statue zoom

Fig. 11. Examples of general drawings modeled as Portuguese pavements

Computing the CVD based on primitive drawing imposes that the result accuracy is limited by the framebuffer resolution. We can use offscreen rendering buffer, but the solution does not scale to large domains. As a consequence, the maximum number of stones is also limited. In our experiments, we have observed that setting the average half side, $\bar{h}$, to a value less than 3.0 degrades the shapes of the generated stones.

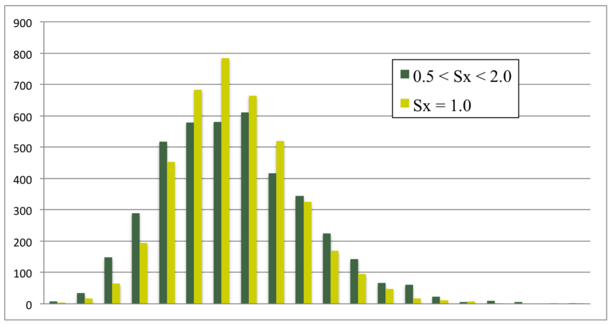


Fig. 12. Histogram of stone areas: a constant scale factor produces more stones with similar size (area)

V. Conclusion

In this paper, we proposed a new method for mosaic modeling. Our goal was to mimic Portuguese pavements, especially the Copacabana beach sidewalk pavement with its waves. The proposed solution does attend the main particularities of such mosaics: the stones completely follow the edges, being mostly represented by irregular quadrilaterals with no particular orientation. Our method has demonstrated to be stable and has been applied to a variety of drawings.

In the current implementation, only the CVD computation is done based on the graphics hardware. As future work, we plan to implement the entire algorithm on the GPU and to investigate a procedure to generate realistic procedural texture in realtime. We also plan to investigate the rendering of Portuguese pavements, taking into account deterioration with time and imperfections. Another natural and straightforward extension is to apply the proposed method to colored images.

Acknowledgment

The first author received financial support from Globo.com to conduct this research. The second author received research grant from CNPq (Brazilian National Research and Development Council).

References

[1] S. Battiato, G. D. Blasi, G. M. Farinella, and G. Gallo, "Digital mosaic frameworks - an overview," *Computer Graphics Forum*, vol. 26, pp. 794–812, 2007.

[2] P. Haeberli, "Paint by numbers: abstract image representations," *SIGGRAPH Comput. Graph.*, vol. 24, no. 4, pp. 207–214, Sep. 1990. [Online]. Available: http://doi.acm.org/10.1145/97880.97902

[3] A. Hausner, "Simulating decorative mosaic," in *Proc. ACM SIGGRAPH '01*, 2001, pp. 573–578.

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

[5] G. Di Blasi and G. Gallo, "Artificial mosaics," *The Visual Computer*, vol. 21, pp. 373–383, 2005, 10.1007/s00371-005-0292-4. [Online]. Available: http://dx.doi.org/10.1007/s00371-005-0292-4

[6] L.-P. Fritzsche, H. Hellwig, S. Hiller, and O. Deussen, "Interactive design of authentic looking mosaics using voronoi structures," in *IN PROC. 2ND INTERNATIONAL SYMPOSIUM ON VORONOI DIAGRAMS IN SCIENCE AND ENGINEERING VD 2005 CONFERENCE (2005*, 2005, pp. 1–11.

[7] K. E. Hoff, III, J. Keyser, M. Lin, D. Manocha, and T. Culver, “Fast computation of generalized voronoi diagrams using graphics hardware,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 277–286. [Online]. Available: http://dx.doi.org/10.1145/311535.311567

[8] D. J., H. T., J. H., and N. T, “A method for creating mosaic images using voronoi dia- grams,” in *Proc Eurographics*, 2002, pp. 341–348.

[9] G. M. Faustino and L. H. de Figueiredo, “Simple adaptive mosaic effects,” in *Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing*, ser. SIBGRAPI ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 315–. [Online]. Available: http://dx.doi.org/10.1109/SIBGRAPI.2005.46

[10] L. Zhang and J. Yu, “Image Mosaics with Irregular Tiling,” in *IEEE International Conference on Computer-Aided Design and Computer Graphics*, 2011.

[11] V. A. d. Passos and M. Walter, “Simulation and rendering of opus palladium 3d mosaics,” in *Proceedings of the 2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*, ser. SIBGRAPI ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 263–269. [Online]. Available: http://dx.doi.org/10.1109/SIBGRAPI.2008.12

[12] M. Jones, J. Baerentzen, and M. Sramek, “3d distance fields: a survey of techniques and applications,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 4, pp. 581 –599, july-aug. 2006.