

Mosaic Animations from Video Inputs

Rafael B. Gomes, Tiago S. Souza, and Bruno M. Carvalho

Departamento de Informática e Matemática Aplicada

Universidade Federal do Rio Grande do Norte

Campus Universitário, S/N, Lagoa Nova

Natal, RN, 59.072-970 - Brazil

rafaelufnr@gmail.com, souza_san@yahoo.com.br, bruno_m.carvalho@yahoo.com

Abstract. Mosaic is a Non-Photorealistic Rendering (NPR) style for simulating the appearance of decorative tile mosaics. To simulate realistic mosaics, a method must emphasize edges in the input image, while placing the tiles in an arrangement to minimize the visible grout (the substrate used to glue the tiles that appears between them). This paper proposes a method for generating mosaic animations from input videos (extending previous works on still image mosaics) that uses a combination of a segmentation algorithm and an optical flow method to enforce temporal coherence in the mosaic videos, thus avoiding that the tiles move back and forth the canvas, a problem known as *swimming*. The result of the segmentation algorithm is used to constrain the result of the optical flow, restricting its computation to the areas detected as being part of a single object. This intra-object coherence scheme is applied to two methods of mosaic rendering, and a technique for adding and removing tiles for one of the mosaic rendering methods is also proposed. Some examples of the renderings produced are shown to illustrate our techniques.

1 Introduction

Non-Photorealistic Rendering (NPR) is a class of techniques defined by what they do not aim, the realistic rendering of artificial scenes. NPR techniques, on the other hand, aim to reproduce artistic techniques renderings, trying to express feelings and moods on the rendered scenes. Another way of defining NPR is that it is the processing of images or videos into artwork, generating images or videos that can have the visual appeal of pieces of art, expressing the visual and emotional characteristics of artistic styles (e.g. brush strokes).

Animation techniques can convey information that cannot be simply captured by shooting a real scene with a video camera. However, such kind of animation is labor intensive and requires a fair amount of artistic skill. NPR techniques can be used to generate highly abstracted animations with little user intervention, thus, making it possible for non-artist users to create their own animations with little effort.

Mosaic is an Non-Photorealistic Rendering (NPR) style for simulating the appearance of decorative tile mosaics. To simulate realistic mosaics, a method must

emphasize edges in the input image, while placing the tiles in an arrangement to minimize the visible grout (the substrate used to glue the tiles that appears between them), i.e., maximizing the area covered by the tiles, as defined initially in [1]. Another characteristic common to most real mosaic styles are that the tiles are convex.

If one wants to generate mosaic animations, he/she has to track the tile locations and enforce that their geometrical relation maintains temporal coherence, i.e., does not suffer from abrupt changes, to avoid discontinuities over time, or swimming, where drawn features can move around the canvas. The method proposed in this paper creates temporally coherent mosaic animations from input videos, extending previous works on still image mosaics. This paper introduces a method for enforcing temporal coherence in mosaic videos that is based on a combination of a segmentation algorithm and an optical flow method. The result of the segmentation algorithm is used to constrain the result of the optical flow, restricting its results to the areas detected as being part of a single object. The main contributions of this paper are the extensions of two still mosaic techniques [2,3] for generating mosaic animations. These extensions include a method for moving tiles in a temporal coherent way, as well as methods for adding and removing tiles.

2 Still Image Mosaics

The generation of artificial still mosaics must follow a few rules if the intent is to generate images similar to common man made mosaics, such as the maximization of the area covered by the tiles, the use of convex tiles, and the emphasizing of edges by orienting the tiles according to the edge orientation.

The use of Voronoi diagrams to generate artificial mosaics is very popular, since they discretize the 2D space into finite convex regions (tiles) and maximize the space covered by the tiles. The first attempt to produce images with a mosaic style effect was proposed by Haeberli [4], that worked by creating random sites for the Voronoi diagram and painting each region with a color sampled from the input image. In order to produce a smoother flow of tiles that follow the edges detected in the input images, Hausner [1] proposed to use a generalization of Centroidal Voronoi diagrams (CVDs), that are Voronoi diagrams that have the additional property that each site is located in the center of mass of its region. The CVDs are calculated using an iterative algorithm that updates the centroid positions and recomputes the Voronoi diagrams until it converges, and it can be implemented in hardware, thus speeding up its execution. The orientations of the tiles are controlled by a direction field that can be created using the Euclidean distance or the Manhattan distance from the edges, if the desired tile shapes are hexagonal or square, respectively.

Dobashi et al. [5] proposed a method where the initial sites are positioned approximately at the centers of an hexagonal mesh, and thus, are approximately centroidal. The sites are then moved to minimize a metric defined based on the color of the pixels contained in each tile and the Voronoi diagram is recom-

puted, thus, representing global features of the input image. In [3], Faustino and Figueiredo proposed an adaptive mosaic method where tiles of different sizes are used, according to the feature size in the input image, according to an image density function.

Recently, Di Blasi and Gallo [2] proposed a mosaic method based on the Distance Transform Map (DTM) to create chains spaced periodically according to a pre-defined tile size, where the DTM is produced based on a guideline (edge) image that encodes the edges to be emphasized in the rendering. The tiles are then placed following these chains. This technique has problems handling noisy images, since the existence of small guidelines (edges) produce small cut tiles and images that do not resemble a typical man made mosaic.

In this work we use Voronoi diagrams and CVDs to create an animated mosaic rendering style, whereas the mosaics obtained from the CVDs are more similar to the traditional man made mosaics. Another problem with the animations produced by using the Voronoi diagrams is that distances between neighbor tiles can vary throughout the animation, generating a distracting effect. We also propose to use a new initial site distribution scheme for computing the CVDs, followed by the use of a Constrained Optical Flow method to maintain intra-object temporal coherence throughout the animation.

3 Animated Mosaics

One of the objectives of NPR techniques for video stylization is to make automatic or semi-automatic procedures that mimic real life artistic styles, thus allowing a user to stylize real movie sequences captured with a camera with little effort when compared to the task of creating an animation from scratch. Video stylization also offers the choice of mixing real movies with stylized objects, rendering with one or more NPR techniques only parts of a movie sequence, leaving the rest of the video sequence intact.

When producing a NPR video from a modeled 3D scene, it is important to maintain temporal coherence, moving the elements of the drawing (e.g. brush strokes) with the surfaces of the objects being drawn, otherwise, these elements stick to the view plane and the animation appears as if it is seen through a textured glass. due to this nature this is referred as the *shower door* effect, named by Meier in [6].

However, if the input for the NPR video is a normal video, it has been reported in the literature that not maintaining temporal coherence incurs in *swimming*, where features of the animation move within the rendered animation. This flickering comes not only from changed objects being rendered with elements that follow the object movement but also from static areas being rendered differently each time.

To solve part of this problem, Litwinowicz [7] introduced a method for maintaining temporal coherence in video sequences stylized using an impressionist style. The method consists of using optical flow tracking movement in the scene and move, add or remove brush strokes from frame to frame. An approach for

coherent rendering of static areas in successive frames was proposed by Hertzmann [8], by detecting areas of change from frame to frame and painting over them, i.e., keeping the brush strokes of the static areas. Intra-object temporal coherence is achieved by warping the brush stroke’s control points using the output of an optical flow method. Wang et al. proposed in [9] a method for creating cartoon animations from video sequences by using a mean shift segmentation algorithm for end-to-end video shot segmentation. After the segmentation is performed, the user specifies constraint points on keyframes of the video shot through a graphical interface. These points are then used for interpolating the region boundaries between keyframes.

The animated mosaic technique described by Smith et al. in [10] proposes a method for moving groups of 2D primitives in a coordinated way, thus, allowing a user to create mosaic animations with temporal coherence. The tiles are geometric shapes that are fitted inside 2D containers (polygons) with the help of the user in the first frame. Then, the system automatically advects the container’s tiles to the other frames, in a way that enforces temporal coherence; a step that can be followed by manual refinement. However, since the method of [10] takes as input an animated scene represented as a collection of polygons, it cannot be directly applied to a real video. An extension of this method was proposed in [11], where a Fast Fourier Transform based method was used to perform effective tile placements, allowing the packing of 3D volumes using temporally repeating animated shapes.

Our method for producing intra-object temporally coherent NPR animations is divided into three parts, the segmentation of the input video, followed by the calculation of the Constrained Optical Flow map, and the rendering of objects using some NPR style (in this case, animated mosaics), which we proceed to describe now. Actually, our method could be used to generate the polygon collection representation needed by the method of [10]. The interactions between the parts mentioned above can be seen in Figure 1.

3.1 Video Segmentation

As mentioned above, the segmentation images are used to delimit the extent of the objects, in other words, the search area for the optical flow algorithm. The system described by Collomosse et al. in [12] uses 2D algorithms for segmenting objects in the frames independently followed by the application of a region association algorithm with an association heuristics. This results in a set of temporally convex objects that can then be rendered.

In this paper, the video shots were treated as a 3D volume and interactively segmented using a variant [13] of the fast fuzzy segmentation algorithm introduced by Carvalho et al. in [14], that was extended for segmenting color 3D volumes. The algorithm works by computing, for every voxel of the 3D volume $I(x, y, z)$ (considering the frames as z slices), a *grade of membership value*, between 0 and 1 to a number of objects in the scene, i.e., a segmentation map $S(x, y, z)$.

The user interaction of the segmentation algorithm is the selection of seed voxels for the objects to be segmented. This interaction allows the user to solve

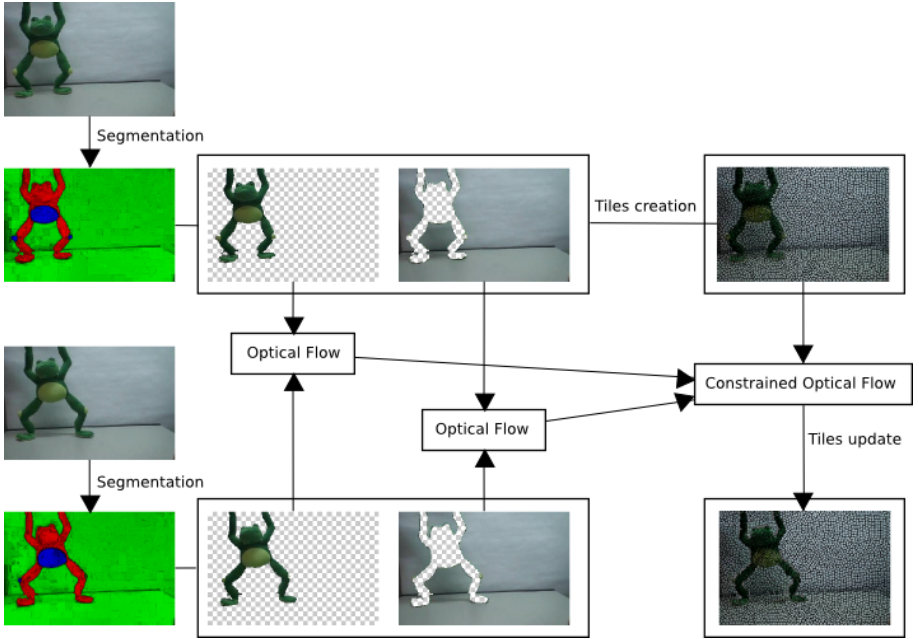


Fig. 1. Diagram showing the interactions between the parts of our method for generating intra-object temporally coherent NPR animations

problems pointed out by Collomosse in [12] as drawbacks for end-to-end 3D segmentation of the video sequence, such as the segmentation of small fast moving objects, gradual shape changes, and texture segmentation, since the user can put seeds throughout the video sequence to capture such object changes.

The fuzzy nature of the segmentation algorithm allows that we render a single object using different styles, according to their grade of membership, e.g., small features inside an object may be detected by their low grade of membership to the surrounding object and be rendered using the original input value.

Here, the objects were segmented based on their color information, but these end-to-end segmentations can be made more robust using not only intensity and color information, but also motion cues, such as the algorithms presented in Galun et al. [15] and Khan and Shah [16], allowing the algorithm to differentiate between foreground and background objects of similar color as one occludes the other.

3.2 Constrained Optical Flow

In video stylization, some authors have used optical flow techniques for enforcing temporal coherence, such as the work of Litwinowicz [7] or the work of Hertzmann [8]. However, the local characteristic of the optical flow techniques and their sensitivity to noisy images somehow limit their applicability. To overcome

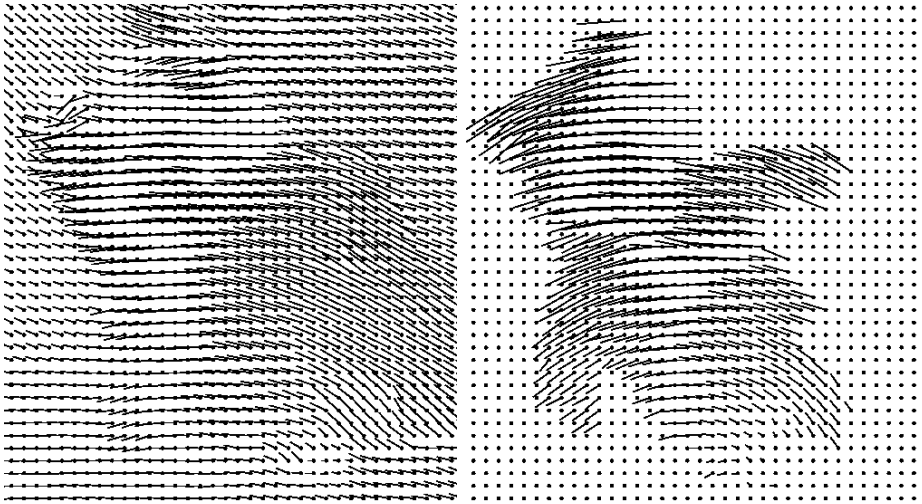


Fig. 2. Application of the optical flow algorithm to two subsequent frames of the Pooh sequence, on the whole image (left) and to the segmented object only (right). Looking at the original sequence, one can see that the Constrained Optical Flow yields better results, specially close to the borders of the Pooh.

those problems, segmentation algorithms have been applied to video shot segmentation to produce end-to-end segmentations that are later used to enforce temporal coherence, as done by Collomosse et al. [12] and Wang et al. [9].

Wang et al. [9] proposed a method where the user selects keyframe points to guide the segmentation, with a typical keyframe interval of 10 to 15 frames, and no intra-object coherence is needed, since the NPR style used are cartoon styles. If sub-regions within an object are needed, the user has to add them using keyframe point selection. In the approach proposed by Collomosse et al. [12], intra-object temporal coherence is achieved by computing an homography, with the strong assumption that the object in question is approximately flat. This may cause severe intra-object distortion in areas with high curvature values.

In this work, we advocate the usage of an optical flow algorithm for enforcing temporal coherence in video NPR sequences, but with the search area for the pixel matching restricted by object boundaries obtained during the segmentation phase. Thus, the optical flow information can be used to enforce intra-object temporal coherence on these sequences. The use of high level knowledge, in the form of a segmented image, provides important information regarding relationship of different objects through time but can also encode information about the type of animation sought by the user. Figure 2 shows two optical flow maps of a frame of the Pooh video sequence.

In order to detect parts of the Pooh object that are moving in adjacent frames, a high value has to be used for the smoothness criterion of Proesmans' algorithm, propagating flow vectors to the background area, even though it is not moving. To use such information would case the background tiles to move unnecessarily.

Furthermore, it can be observed from the input sequence that the Constrained Optical Flow map is much more accurate than the global optical flow map.

The optical flow algorithm chosen for computing intra-object optical flow was the one published in Proesmans et al. [17] because it produces a very dense optical flow map (with one motion estimate per pixel). An evaluation performed by McCane et al. [18] with three complex synthetics scenes and one real scene showed that the algorithm of Proesmans et al. [17] was the only of the evaluated algorithms to produce accurate and consistent flow vectors for every pixel of the image. The algorithm uses a system of 6 non-linear diffusion equations that computes a disparity map and also depth discontinuity map, containing information about occluded parts. This depth discontinuity map may be useful in maintaining the temporal coherence in parts of objects that are occluded for short periods.

The Constrained Optical Flow can be defined as follows: given, for every voxel of the 3D image $I(x, y, z)$ (considering the frames as z slices), a *grade of membership value*, between 0 and 1 to a number of objects in the scene, in the form of a segmentation map $S(x, y, z)$, we have that $S_k(x, y, z) = 1$ if the pixel (x, y) of the z slice belongs to the k th object, and $S_k(x, y, z) = 0$, otherwise.

Based on the membership information of the segmentation, we define the image I_k as

$$I_k(x, y, z) = \begin{cases} I(x, y, z), & \text{if } S_k(x, y, z) = 1; \\ T, & \text{otherwise,} \end{cases} \quad (1)$$

where T is a value outside the range of the images. This ensures that the optical flow is computed only inside a particular object. Thus, the Constrained Optical Flow calculated from two successive frames is given by the union of non-null flow vectors of the calculated Constrained Optical Flow from the individual objects. It is important to note that we do not have to compute the Constrained Optical Flow for all objects, since we can choose not to render an object using a NPR technique, or to render it using a technique that needs only temporal coherence between the borders of objects.

3.3 Rendering

The rendering phase is divided into the definition of the initial distribution, followed by the application of Lloyd's algorithm, and the final rendering of the tiles, that we describe now.

Centroidal Voronoi diagrams tend to fill the space uniformly, creating regions that are approximately regular polygons. In our work, as was done by Hausner [1] and Faustino and Figueiredo [3], we transform a Voronoi diagram obtained from an initial site distribution into a CVD using Lloyd's algorithm [19]. The initial distribution greatly influences the convergence of Lloyd's algorithm, and starting from an initial guess that is approximately centroidal usually requires less site movements and iterations to reach convergence. The initial site distribution can be used to emphasize image characteristics, for example, by using regions of different sizes, specially close to edges, as was done by Faustino and

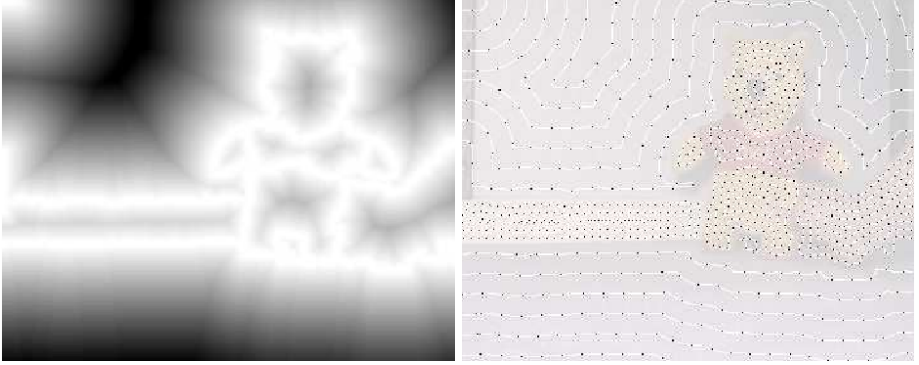


Fig. 3. Distance Transform Matrix (left) and the initial point distribution for the initial frame of the Pooh video input sequence (right). (For visualization purposes, the histogram of the DTM image has been equalized and the gray level have been inverted.)

Figueiredo [3]. On the other hand, successive iterations of Lloyd’s algorithm will tend towards a uniform region distribution, a result that goes against the desired emphasis of some image characteristics. If a close to centroidal distribution is used, Lloyd’s algorithm can be used without substantially affecting the initial non-uniform point distribution.

In our method, we use point chains formed from Distance Transform Matrices (DTM), as done by Di Blasi and Gallo [2] to distribute the tiles. Thus, we can render mosaics using CVDs, as done by Hausner [1] and Faustino and Figueiredo [3], as well as quadrilateral tiles, as done by Di Blasi and Gallo [2]. A DTM is calculated by evaluating at each pixel, its distance from an object border, as can be seen on the left side of Figure 3, where distance zero is white and the farthest pixels are black. Based on the DTM M , the gradient matrix G can be computed by

$$G(x, y) = \arctan \frac{M(x, y + 1) - M(x, y - 1)}{M(x + 1, y) - M(x - 1, y)}, \quad (2)$$

that will be used to determine the tile orientations in the mosaic.

Then the DTM M is used to determine the level line matrix L , computed by

$$L(x, y) = \begin{cases} 1, & \text{if } \text{mod}(M(x, y), 2tSize) = 0; \\ 2, & \text{if } \text{mod}(M(x, y), 2tSize) = tSize; \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where $tSize$ is the tile size. This matrix then determines the lines in which the center of the tiles can be located (pixels x, y such that $L(x, y) = 2$), as can be seen on the right side of 3.

However, here we use their technique to compute an initial site distribution that is approximately centroidal. Figure 3 shows the DTM and initial point distribution of an input video sequence.

However, the method of Di Blasi and Gallo [2] handles only tiles of the same size. This is not the case with our method, since we segment the video sequence into disjoint objects that can have different characteristics associated with them, such as the tile size, emphasizing regions close to borders, as was done in Faustino and Figueiredo [3]. We could even render different objects using different NPR styles, even though this is not done here.

3.4 Adding and Removing Tiles

As objects move closer or further away from the camera, or when new parts of the scene appear in the video, we have to insert or remove new tiles in the animation to maintain a consistent appearance of the tiles, i.e., an homogeneously dense animated mosaic. We now describe a technique we developed to maintain this homogeneous tile packing in animated mosaics.

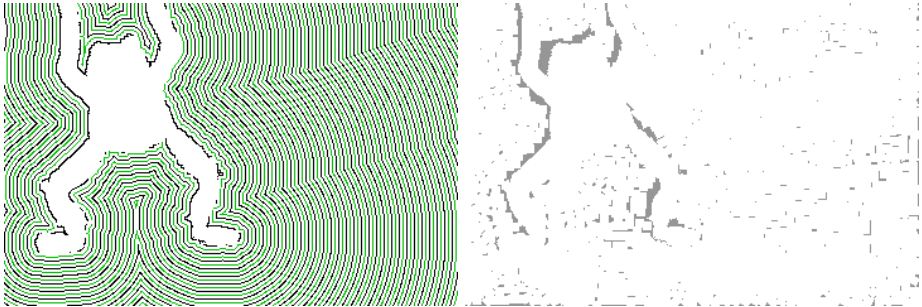


Fig. 4. DTM with the guidelines for tile placement (for the background object) of a frame from the Frog sequence (left) and the areas not covered by tiles of the previous frame moved using the Constrained Optical Flow information (right).

Tile Removal. Tile removal must be used when areas visualized in the previous frame are occluded by the movement of some object in the video or when an object moves further away from the camera. The last case of tile removal happens because the technique of Di Blasi and Gallo [2] uses tiles with the same size, and so, the decrease in area of the object in question, means that less tiles will be used to render it.

In both cases, we use a threshold that specifies the maximal superposition that two tiles can have. The superposition of two tiles appears as if the one on the back has been slightly cut to fit in the area left by the other tiles. Remember that we do have information about the object delineations from the segmentation result. Thus, we render the objects, and compute, based on the Constrained Optical Flow information, which tiles moved to different segmented objects. These tiles, together with the tiles that moved to outside the image are removed and not rendered. As mentioned above, their removal is subject to comparing their intersection area with other segmented object areas or areas outside the frame, to the specified threshold.

Tile Addition. The addition of tiles may be rendered necessary when the area not covered by the tiles grow. This happens when areas not seen in the previous frame appear in the current frame or when no tile is mapped to some area due to an object becoming bigger. In this last case, what happens is that tiles from an object that is moving closer to the camera are moved away from each other, using the Constrained Optical Flow information, and, at some point, the area between them is big enough for a new tile to be rendered. The addition of a tile is done in the following way: working object by object, first we compute the DTM of the object. Then, as done before, we compute the lines in which the center of the tiles can be located. Finally, using a map with 0 where there is a tile and 1 where there is no tile, we insert a new tile, if its intersection with other tiles is smaller than a specified threshold. The maps used in this process can be seen in Figure 4. Playing with the threshold we can achieve more or less tightly packed tiles in areas where the video is changing.

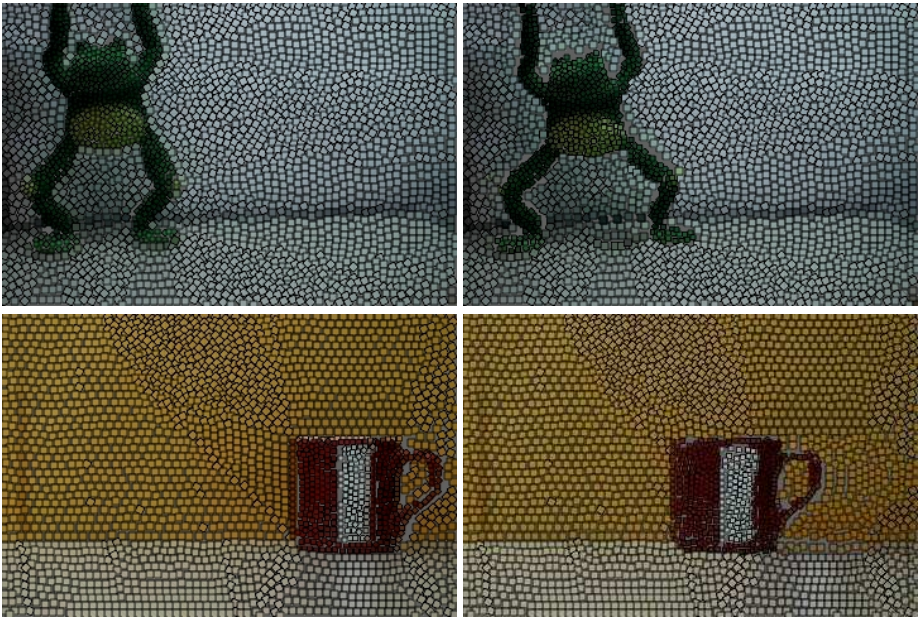


Fig. 5. The 1st and 25th frames of the Frog video, on the top row, were rendered using our techniques for enforcing temporal coherence and for adding and removing tiles. The bottom row shows the 1st and 15th frames of the Mug video, rendered using the same techniques.

4 Experiments

The first and second experiments shown here demonstrate the use of our technique for adding and removing tiles. The top row of Figure 5 shows the first and 22nd frames of the Frog video, where the thresholds set for removing existing

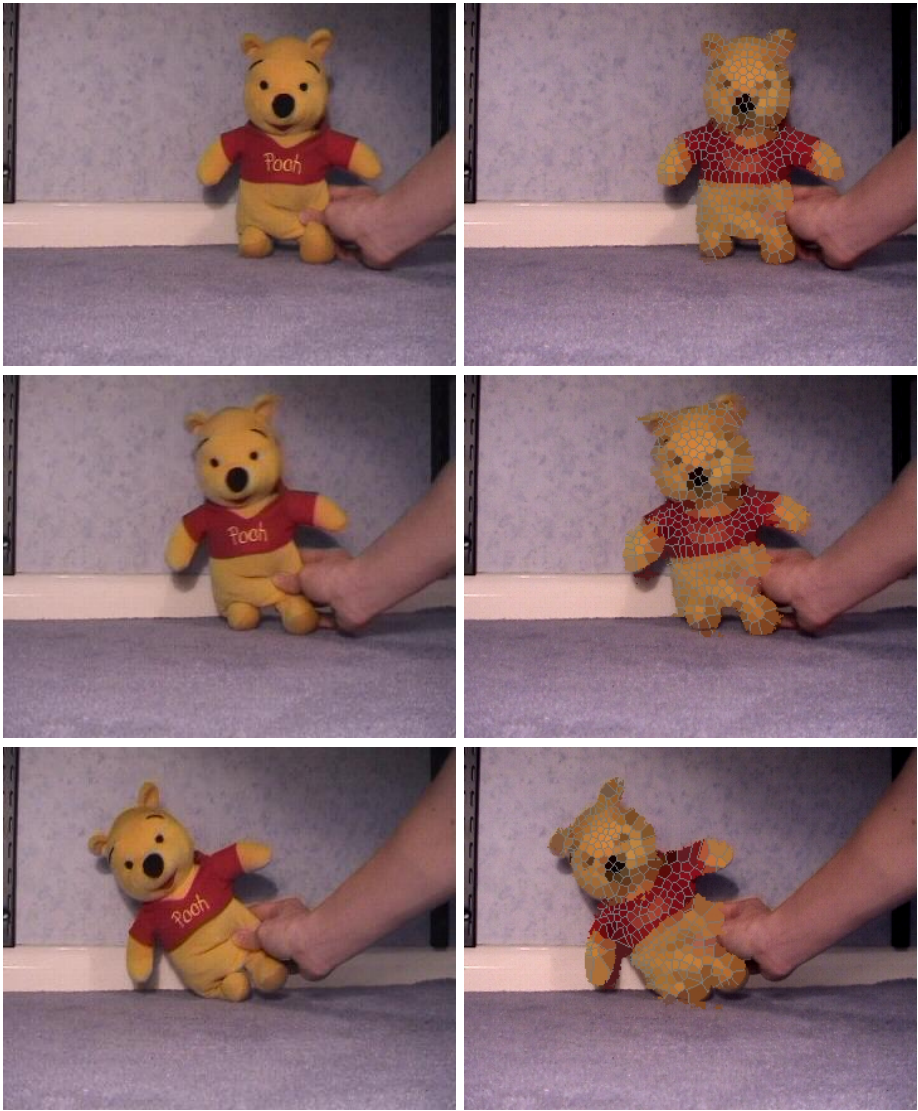


Fig. 6. Three frames of the Pooh input video sequence (left) and the correspondent frames of the mosaic animation (right), where only the Pooh object has been rendered in the mosaic NPR style

tiles and adding new tiles are both 50%. Note that we chose to render the frog with smaller tiles than the background. We do have this flexibility because we segment the video into semantic regions, or temporal objects. As a matter of fact, we can even render different objects using different NPR styles. The bottom row of Figure 5 shows the first and 15th frames of the Mug video, that was

rendered using the same thresholds for tile additions and removals as the Frog video. Note how the tile placements of the background object change very little when comparing both frames.

In the third experiment, shown on Figure 6, we rendered the Pooh object of the 70 frames long Pooh input video sequence as an animated mosaic while rendering the other areas with their original values. This is only possible due to the flexibility allowed by the segmentation of the video sequence end-to-end followed by treating each object as a layer of a video frame. The tile size choice is very important in determining the overall look of the output video, since tiles that are too big will remove important characteristics from the animation (the same is true for still image mosaic). After selecting the initial site distribution performed using the DTM and gradient matrix, Lloyd’s algorithm was run for 10 iterations and the approximated CVD was tracked using the result of the Constrained Optical Flow method proposed here.

It is very important for the success of our method that the segmentation of the objects be of good quality, otherwise, the flexibility of our method turns against us, rendering in an erroneous way parts of the object that were mistakenly segmented. Of course, noisy videos will affect the quality of the Constrained Optical Flow result, even to the point of making it useless. To better handle noisy input videos, a multi-scale approach such as the one proposed by Galun et al. [15], may be useful. However, the segmentation method described here has been successfully used to segment very diverse videos, some of which contained several overlapping objects and moving shadows [20].

5 Conclusion

We presented a method for generating mosaic animations while maintaining intra-object temporal coherence. Our method is based on the use of a segmentation algorithm for segmenting a video shot, followed by the application of an optical flow algorithm that produces a dense flow map, allowing it to be used to move the tiles between successive frames with reduced coherence problems. The segmentation of the video shot into objects, that are treated as different layers in the rendering process also provides many options in the rendering phase, such as the use of tiles of different sizes to emphasize characteristics of the input movie, or the use of completely different NPR styles for different objects.

We also presented a method for adding and removing tiles in mosaic animations, and showed some frames of two mosaic movies generated using our techniques. The user can influence the generation of the addition/removal of tiles by adjusting thresholds for both tasks.

Future work include the use of weighted Voronoi diagrams, allowing new tiles to grow gradually and current tiles to shrink until a minimum size that would trigger their removal, and the addition of mathematical morphology tools to the segmentation program, thus, allowing the user to correct manually small segmentation errors in a post-processing step.

References

1. Hausner, A.: Simulating decorative mosaics. In: Proc. of ACM SIGGRAPH, pp. 207–214. ACM Press, New York (2001)
2. Blasi, G.D., Gallo, G.: Artificial mosaics. *The Vis. Comp.* 21, 373–383 (2005)
3. Faustino, G., Figueiredo, L.: Simple adaptive mosaic effects. In: Proc. of SIBGRAPI, pp. 315–322 (2005)
4. Haeberli, P.: Paint by numbers: Abstract image representations. In: Proc. of ACM SIGGRAPH, pp. 207–214. ACM Press, New York (1990)
5. Dobashi, Y., Haga, T., Johan, H., Nishita, T.: A method for creating mosaic images using Voronoi diagrams. In: Proc. of Eurographics, pp. 341–348 (2002)
6. Meier, B.: Painterly rendering for animation. In: Proc. of ACM SIGGRAPH, pp. 477–484. ACM Press, New York (1996)
7. Litwinowicz, P.: Processing images and video for an impressionist effect. In: Proc. of ACM SIGGRAPH, pp. 407–414. ACM Press, New York (1997)
8. Hertzmann, A., Perlin, K.: Painterly rendering for video and interaction. In: Proc. of NPAR, pp. 7–12 (2000)
9. Wang, J., Xu, Y., Shum, H.-Y., Cohen, M.: Video tooning. *ACM Trans. on Graph.* 23, 574–583 (2004)
10. Smith, K., Liu, Y., Klein, A.: Animosacs. In: Proc. of 2005 ACM SIGGRAPH/Eurograph. SCA, pp. 201–208. ACM Press, New York (2005)
11. Dalal, K., Klein, A.W., Liu, Y., Smith, K.: A spectral approach to NPR packing. In: Proc. of NPAR, pp. 71–78 (2006)
12. Collomosse, J., Rowntree, D., Hall, P.: Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Trans. on Visualiz. and Comp. Graph.* 11, 540–549 (2005)
13. Carvalho, B., Oliveira, L., Silva, G.: Fuzzy segmentation of color video shots. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 402–407. Springer, Heidelberg (2006)
14. Carvalho, B.M., Herman, G.T., Kong, T.Y.: Simultaneous fuzzy segmentation of multiple objects. *Disc. Appl. Math.* 151, 55–77 (2005)
15. Galun, M., Apartsin, A., Basri, R.: Multiscale segmentation by combining motion and intensity cues. In: Proc. of IEEE CVPR, pp. 256–263. IEEE Computer Society Press, Los Alamitos (2005)
16. Khan, S., Shah, M.: Object based segmentation of video using color, motion and spatial information. In: Proc. of IEEE CVPR, vol. 2, pp. 746–751. IEEE Computer Society Press, Los Alamitos (2001)
17. Proesmans, M., Gool, L.V., Pauwels, E., Oosterlinck, A.: Determination of optical flow and its discontinuities using non-linear diffusion. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 2, pp. 295–304. Springer, Heidelberg (1994)
18. McCane, B., Novins, K., Crannitch, D., Galvin, B.: On benchmarking optical flow. *Comp. Vis. and Image Underst.* 84, 126–143 (2001)
19. Lloyd, S.: Least square quantization in PCM. *IEEE Trans. on Inform. Theory* 28, 129–137 (1982)
20. Oliveira, L.: Segmentação fuzzy de imagens e vídeos. Master’s thesis, Universidade Federal do Rio Grande do Norte, Natal, Brazil (2007)