Generating Time-Coherent Animations from Video Data

Javier Villegas and George Legrady

Experimental Visualization Lab,
Media Arts and Technology Department,
University of California Santa Barbara, Santa Barbara CA 93106, USA
jvillegas@umail.ucsb.edu
http://www.mat.ucsb.edu/jvillegas

Abstract. In this paper, a series of techniques for the creation of time coherent animations from a video input is presented. The animations are generated using an analysis-synthesis approach. Information of the scene is extracted using only a 2D RGB image. No markers or depth images are needed. After the analysis, the image is drawn again using the extracted information. To guarantee temporal coherence when redrawing the image, different alternatives have been explored: Interpolation on the parameters' domain and gradient descent parameter update. The different methods are described and illustrated with images.

Keywords: Computer animation, analysis-synthesis, time coherency.

1 Introduction

Music composers have been using analysis synthesis (A/S) techniques for years. In this set of approaches, a signal is decomposed into different parts and then the set of parts is used to rebuild the signal. In between the two A/S processes, the constituent parts of the signal can be manipulated creatively to generate significant alterations of the original sound while keeping its identity. With this strategy, different manipulation of the audio signal are possible: independent pitch and duration modification, dispersion, robotization , whispering or automatic pitch tunning[19]. An (A/S) approach can also be used on images. Some of the art works of Knowlton [11], Silvers [17], Levin [13,12] or Rozin [4] are examples of A/S techniques applied to still images with creative purposes. Figure 1 shows the output of different A/S processes when the input is the picture of a face.

If A/S strategies are extended to moving images simply by independent processing of every frame, coherency problems will occur. Synthesis objects can suddenly appear, disappear or their parameters can change abruptly. A distracting flickering artifact that is often non desirable will be present. Although this paper focuses on finding strategies for temporal coherency on animations created with an A/S process, similar problems had been explored by the academic community interested on generalize the stylized rendering of still images

A.L. Brooks (Ed.): ArtsIT 2011, LNICST 101, pp. 108–117, 2012.

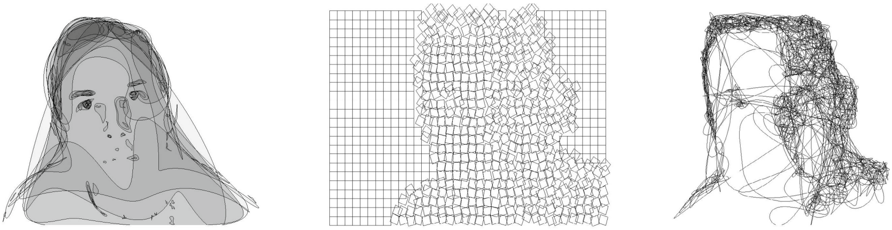

Fig. 1. Different examples of A/S processes applied on a still image

to animations. Bousseau et al [5] presented a technique to create time coherent watercolorizations. They applied deformations to the watercolor textures, using optic flow information from the input sequence. Meier [16], used particles over 3D surfaces to keep track of the position and direction of brush strokes in consecutive frames of pure synthetic non photorealistic scenes. Litwinowicz [14] developed a technique to create hand-painted-looking image animations. He used the edges map of an input image to constrain the length of strokes, then he used optical flow to ensure coherence of the strokes between successive frames. A similar approach but oriented to interactive real-time application was created by Hertzmann and Perlin [10]. Bernard et al had explored the use of dynamic textures and Gabor noise primitives to create time coherent stylizations [1,2]. Finally, in Animosaics [18], Smith, Liu and Klein explored rules for the smooth motion of mosaic tiles in animated mosaics. They pursued not only coherence between individual elements, but cohesion in the movement of groups of tiles.

This approach differs radically from the previously mentioned works. The main interest is not to recreate the look of hand painted images or artistic stiles on video sequences but to explore the narrative possibilities of A/S approaches on video signals with temporal coherence. Next two sections will show two different approaches to generate time coherent animations from video data. First, an approach based on the matching of closer objects in consecutive frames is presented. After the matching, the trajectory of the objects can be interpolated or smoothed. This approach possess some similarities with well known problems like the tracking of partials on audio signals with algorithms like the McAulay Quatieri[15] and the general assignment problem in combinatorial optimization [7]. Next we explore a gradient based approach where local rules followed by the synthesis elements recreate a global image. This strategy resembles works on generative computer art. On section 4, results are summarized and some possibilities for future explorations are presented.

2 Correspondences on the Parameters' Domain

Temporal coherence is lost when frames are analyzed independently because the parameters of objects in adjacent frames can change abruptly. Even more, the number of synthesis objects can be totally different in consecutive frames. This rapid appearing and disappearing of objects destroys the effect of motion at the local level and generates a disturbing popping artifact at a global scale. In order to create a coherent animation, the objects on consecutive frames have to be paired. To define this problem as a general linear assignment problem [7] the number of objects have to be the same on every frame. This obstacle is solved by allowing objects of null size (invisible objects) to exist on the less populated frames. This strategy is similar to the one followed in the seminal work of McAulay-Quatieri [15] to handle the death and birth of sinusoidal partials on adjacent audio frames. In their algorithm, they allowed sinusoidal partials of zero amplitude to be matched in previous and subsequent frames. With this consideration the object matching problem can be stated as a linear assignment problem as follows:

Given a cost matrix $C(m, m+1)$ of size $n \times n$ with elements C_{ij} that represents the euclidean distance between the parameters of every object i on frame m to every object j on frame $m+1$. We want to find the assignment matrix $X(m)$ with elements X_{ij} defined as:

$$X_{ij} = \begin{cases} 1, & \text{if object } i \text{ on frame } m \text{ is matched} \\ & \text{ to object } j \text{ on frame } m+1; \\ 0, & \text{Otherwise} \end{cases}$$

Such that the sum:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} \ . \quad (1)$$

is minimized.

Subject to:

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad (i = 1.2...n) \ .$$

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad (j = 1.2...n) \ .$$

$$X_{ij} \in \{0, 1\} \qquad (i, j = 1.2...n) \ .$$

This is a standard optimization problem known as the assignment problem, many alternatives for solving this problem efficiently can be found on the literature [7]. After solving this assignment problem a time coherent animation can be produced in different ways:

- Smoothing the trajectories. This creates a trade-off between continuity and accuracy of representation.

– Interpolation. New frames can be generated by interpolating the parameters between the matched objects. A good representation accuracy can be accomplished but the frame rate has to be increased.

Figure 2 shows four frames of a video created with elliptic regions. The ellipse in red color is an example of a matched object.

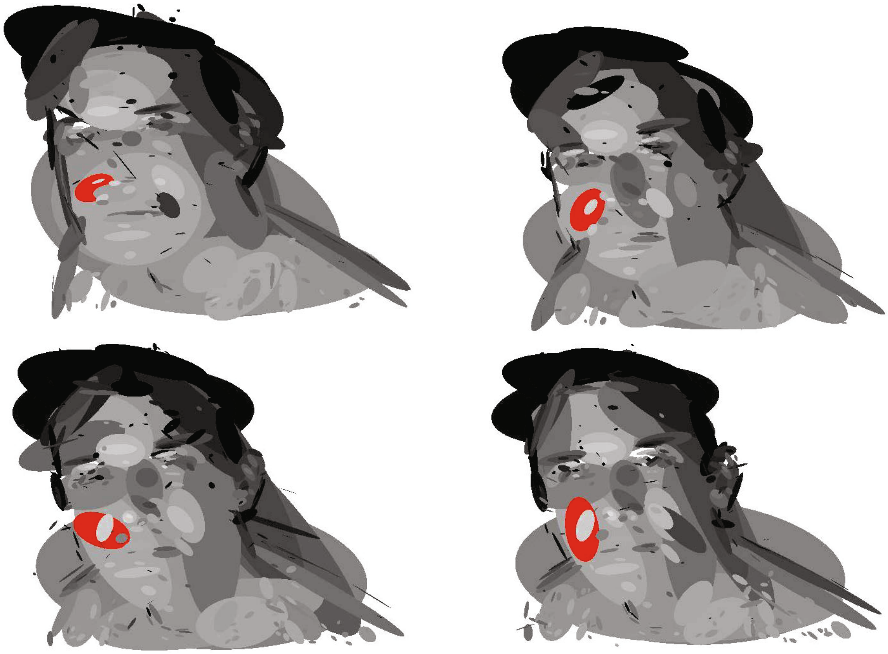

Fig. 2. Four frames of a synthetic video showing in red an object as it was matched

3 Gradient Based Approach

Another alternative to guarantee temporal coherence is to define a set of rules on the synthesis elements. This set of rules will determine the local behavior of the synthesis objects. Global constraints on the object motion are determined by the input image from video data. This image is used to create a surface. The gradient of this surface is used to generate a vector field of forces that will affect the motion of the synthesis elements. Figure 3 shows a general diagram of this approach.

This approach resembles in some way the practice of generative art [9], where a set of rules is defined over a collection of elements that then behave with some degree of autonomy.

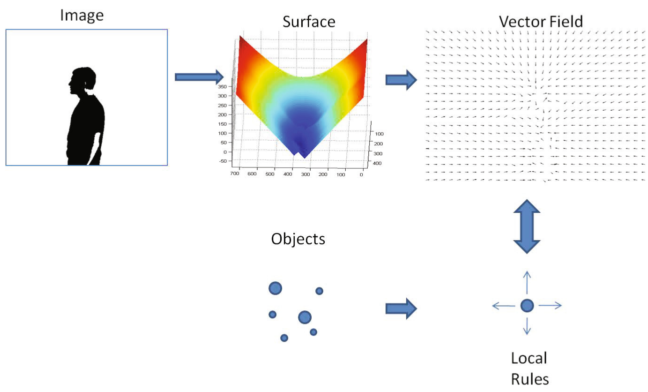

Fig. 3. The generative approach to time coherent animations

3.1 Example 1. A Dynamic Mesh Grid

Figure 4 shows a vector field, in this case generated directly form the gradient of a grayscale image. This vector field is used to attract the nodes of a grid to the dark areas of the input image and repel them from the bright ones. The resulting image is a grid that reassembles the video input. See Figure 5. Internal forces of the mesh such as tension or drag can be modified to produce different visual results.

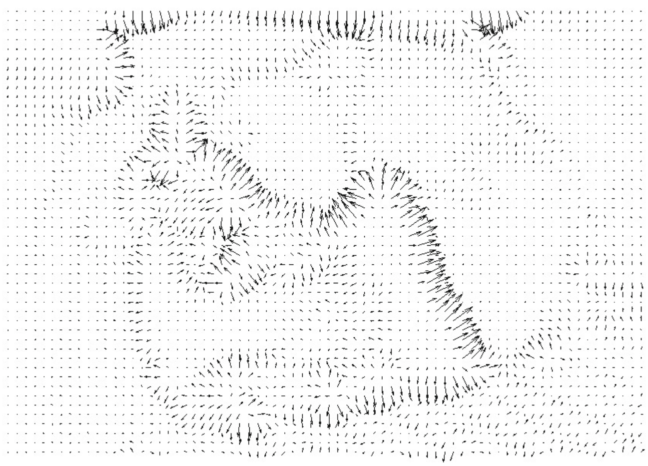

Fig. 4. The vector field obtained with the gradient of a grayscale image), and the mesh grid (Right)

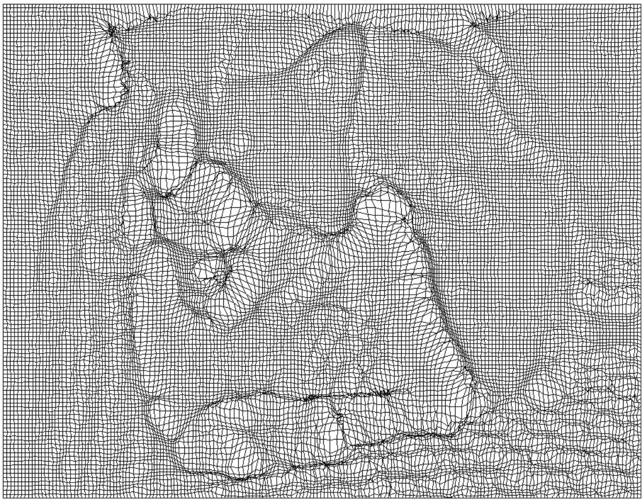

Fig. 5. Mesh grid generated with the vector field

3.2 Example 2. Circle Packing

Figure 6 shows a sequence of frames, where a set of circles of two different sizes are getting attracted by the black areas of a binary video. In order to produce this attraction, the black and white image is used to generate a surface by means of the distance transform. If we define the black pixels of the binary image I, as foreground F and the white ones as background B, then the value of a pixel p on the distance transform image is defined by:

$$Dt\left(p\right) = \underset{q \in F}{min}\left(d\left(p,q\right)\right) \ . \tag{2}$$

A surface is then calculated as:

$$S\left(p\right) = Dt\left(p\right) - \bar{D}t\left(p\right) \ . \tag{3}$$

Where $\bar{D}t\left(p\right)$ is the distance transform of the logic complement of image I evaluated at pixel p. The gradient of this surface combined with a circle packing algorithm will determine the motion of every circle.

3.3 Example 3. Coherent Straight Lines

Almost the same set of steps used on the previous example can be applied on a different domain. For example to recreate the input image with straight lines moving with time coherency the following strategy can be used:

– Obtain the edge map of the image.

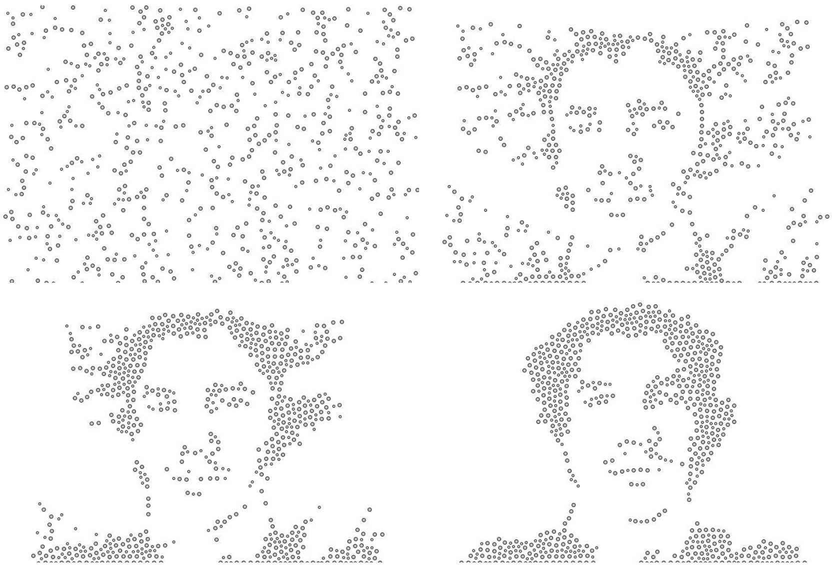

Fig. 6. Some frames of a gradient based animation. The circles started at random position and then they move according to a set of rules and constrained by a surface created with the input image.

– Calculate the Hough transform using the parametrization from [8].

$$r(\theta) = x\cos(\theta) + y\sin(\theta)\ . \tag{4}$$

(x, y) are the coordinates of pixels that belong to an edge.

– Low pass filter the Hough plane image, see left of Figure 7.

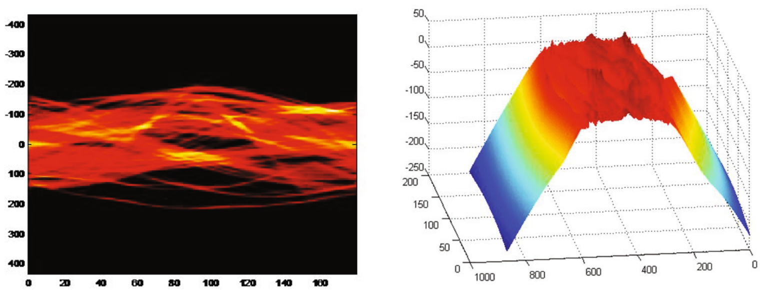

Fig. 7. Left - The low pass filtered version of the Hough transform of an edge image. Right - The surface created applying the distance transform to the Hough plane.

– Get a black and white image by thresholding the resulting image from the previous step.
– Generate a surface using the distance transform as shown in equations 2, 3. See right of Figure 7.
– Generate set of random points on Hough space uniformly distributed. Figure 8 (Left).

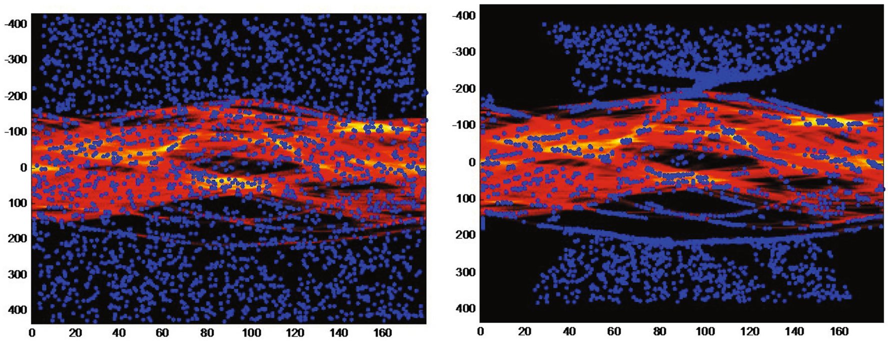

Fig. 8. Left - Random points uniformly distributed on the Hough plane. Right - Dots in the Hough space moving in direction of the peaks.

– Update the position of every point using the gradient of the distance transform and the circle packing algorithm to avoid point collisions. Figure 8 (Right).
– Convert points on Hough space back to lines on the original domain. Figure 9

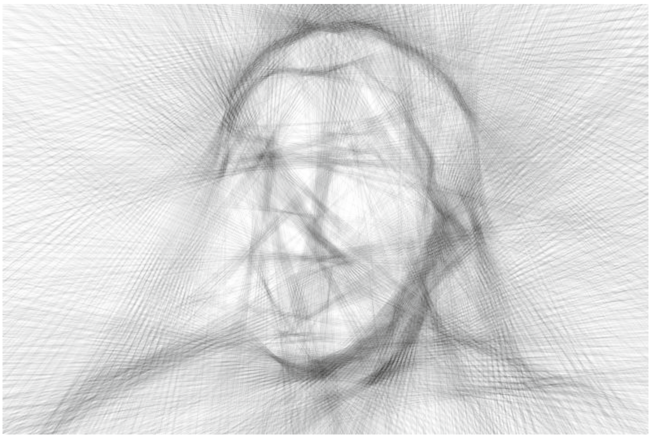

Fig. 9. The image transformed back to the original domain. Every point on Figure 8-Right is now a line.

4 Conclusions

We presented here two separate approaches to generate time coherent animation from video data using analysis and synthesis techniques. The first one uses the matching information of objects in adjacent frames to create smooth transitions or to generate interpolated objects in new in-between frames. This approach still presents a lot of room for exploration, for example: Modifying the cost matrix with direction information to penalize matchings that would produce motion patterns far from the average or fluctuating trajectories. Using optical flow information as preferred direction. Minimizing the maximum error instead of the squared sum (Eq 1) in the matching stage (this variation is known as the bottleneck assignment problem [7]).

The second technique uses a gradient based approach creating a surface in which the synthesis objects can move smoothly. The objects have to be aware of themselves so they do not sink together into a local minimum, that is the reason why this strategy is always combined together with a circle packing algorithm. It was showed that this approach can be extended to parametric curves like straight lines, after a preliminary domain transformation as the Hough transform. Natural extension of this approach include the creation of time coherent animations with different parametric curves like circles, ellipses or parabolas. Future work includes the exploration of synthesis techniques based on the result of well know video tracking techniques as the continuously adaptive mean shift algorithm [6] for region based synthesis and active contours[3] for parametric curve based synthesis.

References

1. Bénard, P., Bousseau, A., Thollot, J.: Dynamic solid textures for real-time coherent stylization. In: ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), pp. 121–127. ACM, Boston (2009)
2. Bénard, P., Lagae, A., Vangorp, P., Lefebvre, S., Drettakis, G., Thollot, J.: A dynamic noise primitive for coherent stylization. Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering 2010) 29(4), 1497–1506 (2010)
3. Blake, A., Isard, M., et al.: Active contours, vol. 2. Springer, London (1998)
4. Bolter, J.D., Gromala, D.: Windows and Mirrors: Interaction Design, Digital Art, and the Myth of Transparency. The MIT Press (October 2005)
5. Bousseau, A., Neyret, F., Thollot, J., Salesin, D.: Video watercolorization using bidirectional texture advection. ACM Trans. Graph. 26 (July 2007)
6. Bradski, G.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal (1998)
7. Burkard, R., Dell'Amico, M., Martello, S.: Assignment problems. Society for Industrial Mathematics (2009)
8. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM 15, 11–15 (1972)
9. Galanter, P.: What is generative art? complexity theory as a context for art theory (2003), `http://philipgalanter.com/downloads/ga2003_what_is_genart.pdf`

10. Hertzmann, A., Perlin, K.: Painterly rendering for video and interaction. In: NPAR 2000, pp. 7–12 (2000)
11. Knowlton. Knowlton mosaics - portraits by computer assisted art pioneer ken knowlton, http://www.knowltonmosaics.com/
12. Levin, G.: Floccular portraits - interactive art by golan levin and collaborators, http://www.flong.com/projects/floccugraph/
13. Levin, G.: Segmentation and symptom - interactive art by golan levin and collaborators, http://www.flong.com/projects/zoo/
14. Litwinowicz, P.: Processing images and video for an impressionist effect. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, pp. 407–414. ACM Press/Addison-Wesley Publishing Co., New York, USA (1997)
15. McAulay, R., Quatieri, T.: Speech analysis/synthesis based on a sinusoidal representation. IEEE Transactions on Acoustics, Speech and Signal Processing 34(4), 744–754 (1986)
16. Meier, B.J.: Painterly rendering for animation. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, pp. 477–484. ACM, New York (1996)
17. Silvers, R.: Robert silvers (2003), http://www.photomosaic.com/
18. Smith, K., Liu, Y., Klein, A.: Animosaics. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2005, pp. 201–208. ACM, New York (2005)
19. Zölzer, U., Amatriain, X., Arfib, D., Bonada, J., Poli, G.D., Dutilleux, P., Evangelista, G., Keiler, F., Loscos, A., Rocchesso, D., Sandler, M., Serra, X., Todoroff, T.: DAFX:Digital Audio Effects, 1st edn. Wiley (May 2002)