Vectorized Painting with Temporal Diffusion Curves
基于时间扩散曲线的矢量绘图

侯飞

IEEE Transactions on Visualization and Computer Graphics, accepted

18610050238

Fig. 1. (1)-(4) The painting session of a still-life artwork and (5)-(6) the zoom-in views of the final result. Our system recreates a very similar workflow to the real-life painting with brushes and papers and remains sharp when zoomed in significantly due to the nature of vector graphics.

Although these digital painting systems can produce good results, they all have common problems. On one hand, these systems use discretized basis in computation and final results, making it nontrivial to change resolution once the painting session begins. On the other hand, the discrete representation makes editing each individual stroke on the fly impractical without re-simulation. Vector graphics and vector painting systems are developed to overcome the above drawbacks. As the underlying representation, vector graphics have many advantages over the discretized basis, including the nature of being resolution independent, the ability for stroke editing and the computational efficiency. Traditional vector graphics are generated from rasterized images using meshes [1], but they lack the fundamental ability for editing. Later, Orzan et al. [2] proposed Diffusion Curve (DC) which solves the Laplace’s equation using curves as the boundary conditions for color spreading. Although this technique excels in producing high-quality results, the DC images are composed of boundary curves rather than strokes, which inevitably contradicts the human intuition and painting habits, see Figure 2. Moreover, the intersection of DCs results in artifacts, which restrains the flexibility of DC. Recently, DiVerdi et al. [3] put forward a vector watercolor painting engine using procedural stroke configurations, bringing great realism in the creation of the digital artworks.

This paper presents a more generic digital painting measure through a novel vector model for strokes. We first propose a new model of vector graphics, Temporal Diffusion Curve (TDC), which represents not only the graphics but also their evolution over time using continuous functions. This new model is piece-wisely parameterized and inherently supports random-access solving in real-time. The TDCs represent strokes similar to human habits of painting and they are able to intersect and overlap each other as usual strokes. Therefore, it is very suitable for modeling strokes. Meanwhile, we devise a procedural model for processing TDC strokes to realize richer visual effects, including smooth paint diffusion, irregular paint scattering, interstroke color mixing, etc. Based on these, we build a painting system of great potential. Specifically, our method has realtime performance regardless of the rendering resolution, provides straightforward editing possibilities on strokes both at runtime and afterward, and delivers various stroke effects for art production of multiple genres. In contrast to the former DC which solves the static Laplace’s equation for color spreading, our method integrates the temporal heat equation instead with TDCs being the diffusion source. More concretely, we find the closed-form solution of the heat equation by using Fourier transform and only compute the numerical density right before the procedural stroke processing. In this way, the painting can use evolving strokes as the basic primitives as opposed to the counterintuitive motionless boundary curves in DC images, see Figure 2 for details.

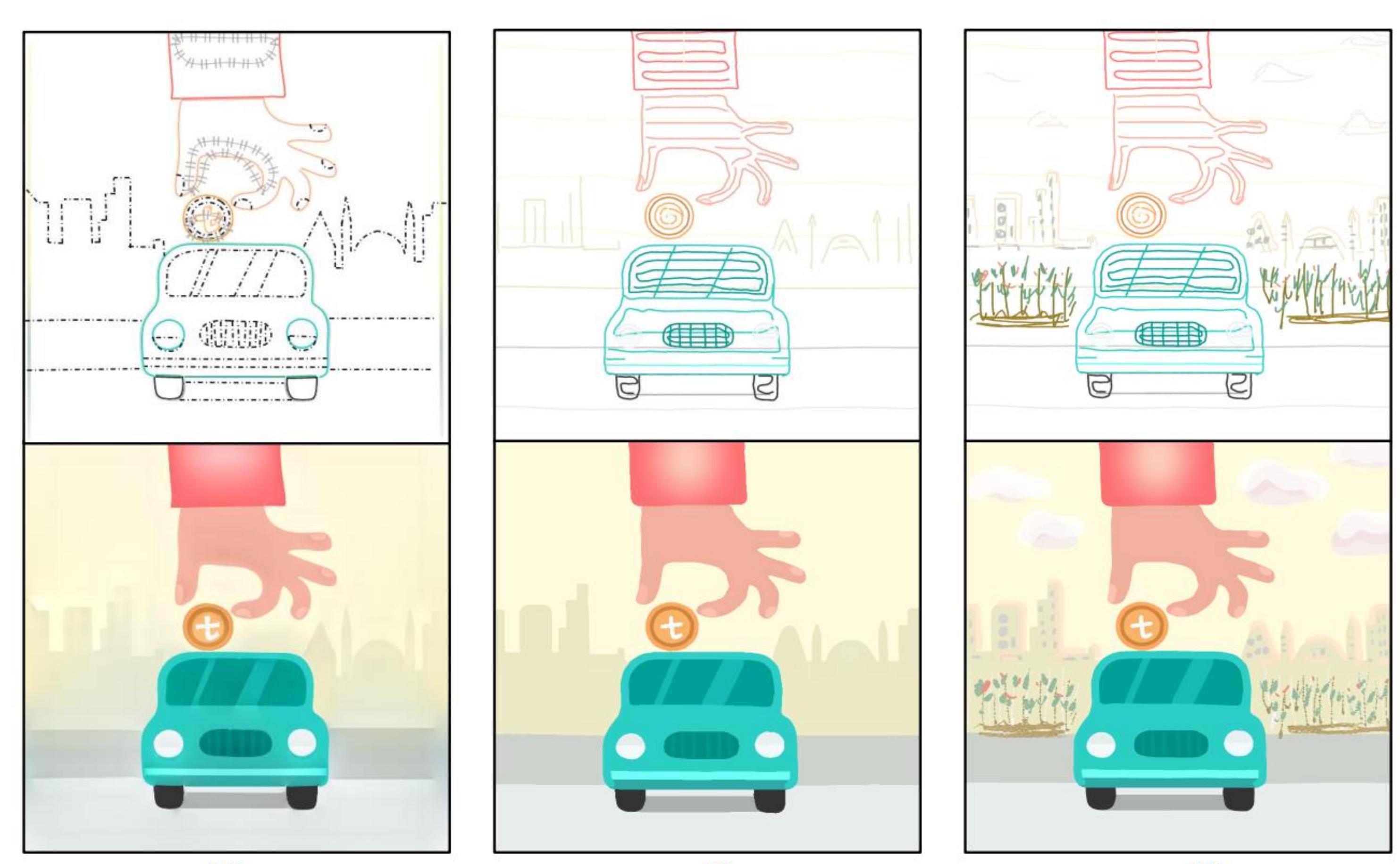

Figure 1. Comparison of inputs and results among Diffusion Curve (PVG) [4] (A), our TDC method (B) and our TDC result with extra details (C). The DC method uses region boundaries and auxiliary curves to determine the color and its variation. In contrast, TDC models the strokes directly, which is much more consistent with the real-life painting experience using brushes and papers. Moreover, it is perfectly feasible to recreate the DC results with TDC. Nevertheless, some results of our method, including the canvas texture, crossing strokes, color layers, and the temporal evolution, are not trivial to implement with DCs.

[1] T. Xia, B. Liao, and Y. Yu, “Patch-based image vectorization with automatic curvilinear feature alignment,” ACM Transactions on Graphics (TOG), vol. 28, no. 5, p. 115, 2009.
[2] A. Orzan, A. Bousseau, H. Winnemoller, P. Barla, J. Thollot, and ¨ D. Salesin, “Diffusion curves: a vector representation for smoothshaded images,” in ACM Transactions on Graphics (TOG), vol. 27, no. 3. ACM, 2008, p. 92.
[3] S. Diverdi, A. Krishnaswamy, R. Mech, and D. Ito, “A lightweight, procedural, vector watercolor painting engine,” in ACM SIGGRAPH Symposium on Interactive 3d Graphics and Games, 2012, pp. 63–70.
[4] F. Hou, Q. Sun, Z. Fang, Y.-J. Liu, S.-M. Hu, A. Hao, H. Qin, and Y. He, “Poisson vector graphics (PVG),” IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 2. pp. 1361-1371, 2020.