1Y-06

Image and Object Space Localized Stylization Method for View-Dependent Lines Extracted from 3D Models

Luis Cardona † Suguru Saito †† †
†Tokyo Institute of Technology ††Ochanomizu University

1 Introduction

Non-Photorealistic Rendering (NPR) focuses on how to imitate the principles of abstraction used by artists. This paper proposes a NPR method for line drawings that enables the user to locally stylize view-dependent lines extracted from 3D models. Previous stylization systems [1, 2] are useful to quickly define the style of an entire scene but lack control over local stylization of view-dependent lines. A comparison with our previous work show that the proposed system in this paper achieves an improvement in the expressiveness of line drawings for both static and animated models.

2 Input phase

2.1 Stroke registration

We provide an user interface to easily apply a wide range of stroke styles to the lines extracted from a 3D model (Fig. 1). The user can customize style properties such as width, color, texture and add line offsets by sketching an example stroke. For each stroke drawn, the system stores its location on the underlying surface along with its style properties. We refer to this stylization data as *registered strokes*. The user can change the camera position in order to input new strokes for lines appearing at different view directions. This process is repeated until the desired style is achieved for all viewpoints.

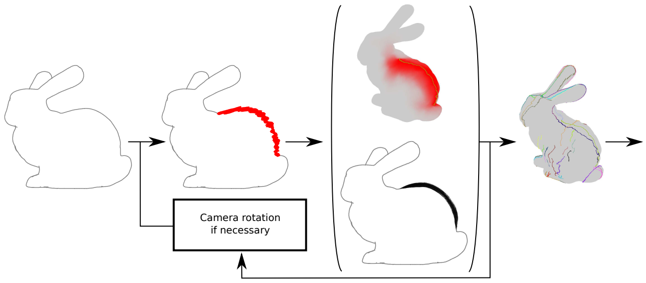

Figure 1: Interactive stroke registration. From left to right: the initial state without stylization; the user traces over a line; the style properties, stroke surface location and area of influence are registered; all *registered strokes* after drawing from multiple viewpoints.

2.2 Areas of influence

We define *areas of influence* as surface regions that delimit the locations where *registered strokes* can be matched to current frame lines. The *area of influence* of a *registered stroke* is generated by diffusion over the surface. When the viewpoint is modified, one problem we face is that while a contour line moves smoothly in image-space, it may move abruptly over the surface. In object-space, the distance on surface that the contour line traverses is inversely proportional to the radial curvature (i.e. the curvature in the view direction) of the underlying surface. Even for small changes in the view direction, the contour line may move long distances when the underlying surface has low radial curvature (e.g. planar surfaces). To solve this problem, we propose a diffusion algorithm that accentuates the contribution of vertices close in screen coordinates. The resulting diffusion is indirectly dependent on the radial curvature and spreads faster in the view direction as shown in Fig. 2. We define a Laplacian approximation with weights equal to the inverse of the lengths of the projected edges into the image plane:

$$\Delta I_i = \sum_{j \in i*} w_{ij}(I_i - I_j) \tag{1}$$

where $i*$ is the set of neighbors of vertex i, I_i and I_j are the influence values at vertices i and j and

$$w_{ij} = \frac{1}{|q_i - q_j|} \tag{2}$$

where q_i and q_j are the projected vertices i and j. The diffusion process is done iteratively using the following conditional explicit Euler scheme:

$$I_i^{n+1} = \begin{cases} 1 & \text{if } i \in T_{RS} \\ 0 & \text{if } i \in T_L \text{ and } i \notin T_{L_{RS}} \\ I_i^n + \lambda \Delta I_i^n & \text{otherwise} \end{cases} \tag{3}$$

where T_{RS} is the set of triangles containing feature line segments that belong to the registered stroke RS, T_L is the set of triangles containing all line segments in the same view when RS was registered, $T_{L_{RS}}$ is the set of triangles containing line segments that belong to the feature line connected to RS and λ is a constant such that $0 < \lambda \leq 1$. Setting the boundary conditions to 0 can be optionally omitted.

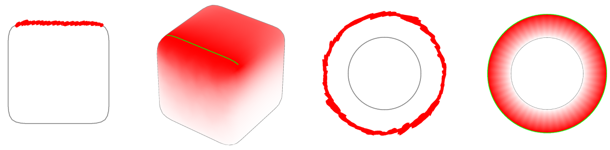

Figure 2: Image-space weighted diffusion. The influence values spread faster over the top side the cube (left). Boundary conditions set to 0 for the inner ring contour of a torus (right).

3 Rendering phase

Our rendering pipeline takes as input a 3D model as well as the *registered strokes* and corresponding *areas of influence*. View-dependent feature lines such as occluding contours [3] and suggestive contours [4] are extracted from the 3D model. These feature lines are then rendered as a binary image B that is used as input to build an attraction field. This attraction field leads 2D polylines generated from the *registered strokes* to current frame contours. The *areas of influence* explained in the previous section serve the purpose of discarding obvious incorrect matches which are consequence of local occlusion and usually occur at surface locations far away from the *registered stroke*. For each polyline successfully matched to current frame feature lines, a stroke is rendered as a triangle strip with the unique style properties stored in the correpонding *registered stroke*.

3.1 Line matching

We propose an image-space line matching method that attracts 2D polylines to the current frame feature lines (Fig. 5). These matching polylines are generated as active contours ("snakes") [5] with sample points $s = (x, y)$ created along the projection of their corresponding *registered stroke* into the image plane. We update all sample points of the active contour by minimizing an energy function composed by an internal and an external energy using Euler-Lagrange integration method. This update process is repeated until the active contour converges or a maximum number of iterations is exceeded. We use the most common internal energy definition in terms of first and second derivatives to minimize length and curvature. In contrast to methods that track lines frame by frame to achieve temporal coherence [6, 7],

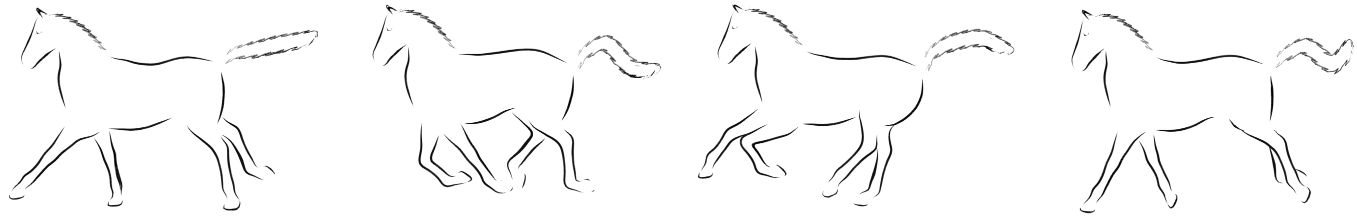

Figure 3: Localized Stylization using different brush textures and line offsets for an animated model.

Figure 4: Comparison of our previous method (left) with the proposed method in this paper (right).

our line matching algorithm needs to lead the matching polylines from regions relatively far away from current frame feature lines. Consequently, we replace the common external energy gradient by an attraction field A that is define for all points on the image that are located over the 3D mesh as well as the feature lines:

$$A(p) = \begin{cases} A_l(p) & \text{if } ||A_l(p)|| > 0 \text{ and } B(p) = 1 \\ A_n(p) & \text{otherwise} \end{cases} \quad (4)$$

where $p = (x, y)$, A_l is the gradient of the result of filtering the binary image B with a Gaussian kernel such that $A_l = \nabla(G_\sigma * B)$ and A_n is the projection of surface normals into image-space. Although the projected normals field A_n provides almost full coverage, line matching lines would not stop at internal contours. By combining A_n with the contour attraction field A_l, the proposed attraction field A provides a path to current frame contours from points far away while being attracted to both external and internal lines. For each *registered stroke*, A is scaled by the values stored at its *area of influence* in order to prevent obvious mismatches.

4 Results

Our system provides an interactive way to locally stylize feature lines extracted from 3D models. Specifying the style of feature lines for models such as the ones shown in Fig. 3 and 4 can be achieved in less than 30 minutes and only needs to be done from a small number of viewpoints. In our previous work [8], the terminal points of strokes were limited to inflection points but the proposed system in this paper enables the user to decide which parts of the feature lines will be drawn. These results also show how strokes with different styles can be used simultaneously on the same model in order to convey additional information such as material properties or feature sharpness.

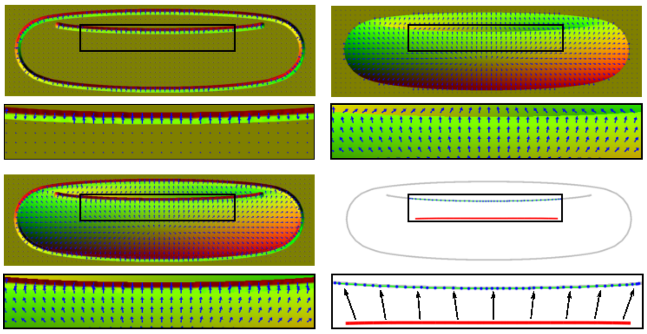

Figure 5: Line matching. A_l is defined only near external and internal feature lines (top left). A_n is defined at almost all points inside the 3D model (top right). The combined attraction field (bottom left) leads the 2D polylines from the registered stroke (red) to current frame feature lines (bottom right).

5 Conclusion and Future work

The proposed method allows the preservation of style properties for nearby viewpoints by matching *registered strokes* to current frame feature lines. Our line matching method is performed in image-space and consequently is robust to changes in connectivity of the object-space lines. The combined attraction field leads the matching polylines to current frame external and internal feature lines far away from the *registered strokes* and the *areas of influence* prevent obvious mismatches. However, complex shapes may produce mismatches when the current viewpoint is too far away from the viewpoint the stroke was registered. Consequently, future works may use additional information from the 3D surface to improve the accuracy of the line matching.

References

[1] Kalnins Robert et al. Wysiwyg npr: drawing strokes directly on 3d models. *ACM Trans. Graph.*, 21(3):755–762, July 2002.

[2] Grabli Stéphane et al. Programmable rendering of line drawing from 3d scenes. *ACM Trans. Graph.*, 29(2):18:1–18:20, April 2010.

[3] Koenderink J.J. et al. What does the occluding contour tell us about solid shape. *Perception*, 13(3):321–330, 1984.

[4] DeCarlo Doug et al. Suggestive contours for conveying shape. *ACM Trans. Graph.*, 22(3):848–855, July 2003.

[5] Kass Michael et al. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[6] Kalnins Robert et al. Coherent stylized silhouettes. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 856–861. ACM, 2003.

[7] Bénard Pierre et al. Active strokes: coherent line stylization for animated 3d models. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, NPAR '12, pages 37–46, 2012.

[8] Cardona Luis et al. Customizable strokes for localized stylization of view-dependent lines extracted from 3d models. In *Cyberworlds (CW), 2013 International Conference on*, pages 140–146, Oct 2013.