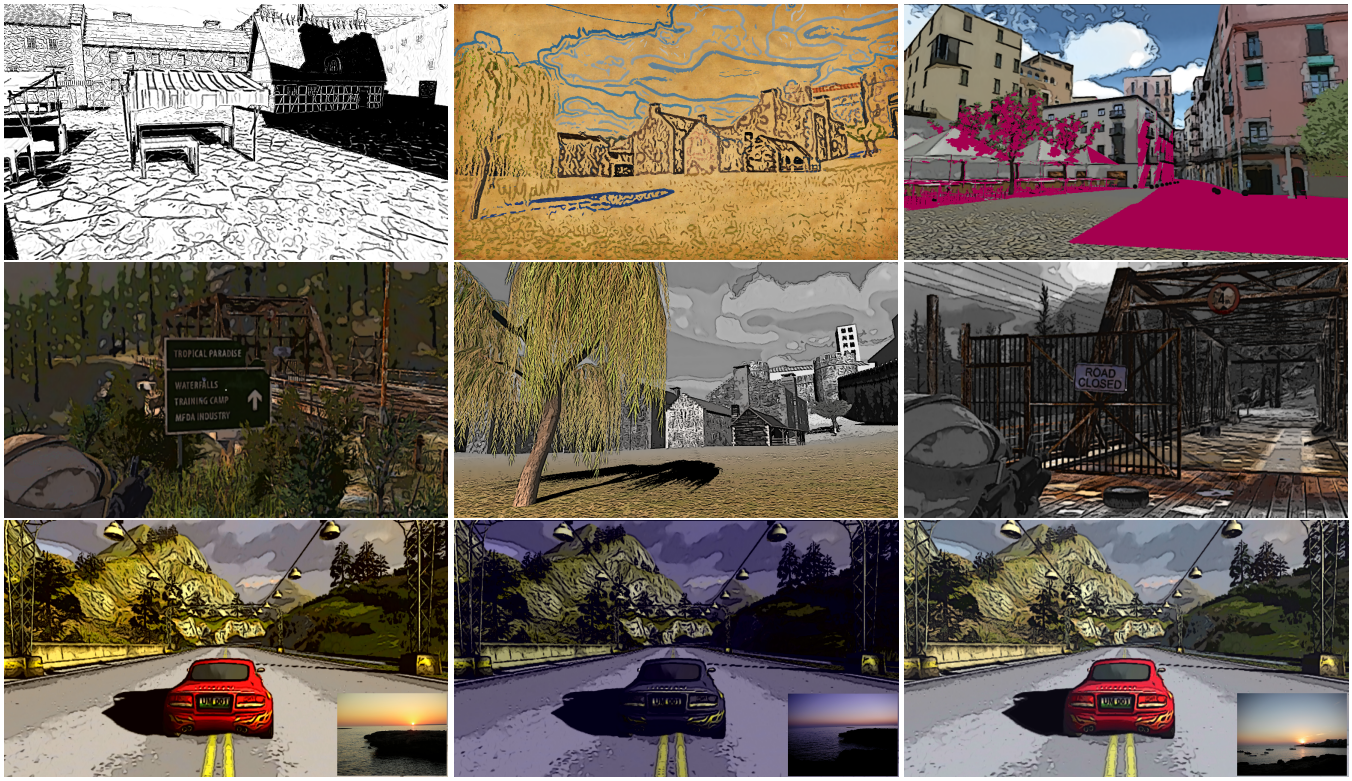
Post-Processing NPR Effects for Video Games: a Case Study

Milán Magdics[1,2], Catherine Sauvaget, Rubén J. Garcia[1], Mateu Sbert[1]

[1]University of Girona, [2]Budapest University of Technology and Economics



(A) (B) (C)

Figure 1: *Example effects. Upper row: (A) Comic-style rendering with edge detection, depth-based edge thickness and black shadows, (B) Edge detection, edge color is set by the color buffer, (C) Colored shadows. Middle row: (A) Abstraction level varying by depth in the Unity Bootcamp demo, (B) Desaturation with smooth transition, (C) Focusing on a specific depth level using color saturation in the Unity Bootcamp demo. Lower row: color stylization by example in the Unity Car Tutorial.*

1 Motivation

The proper *style* of a rendered image or image stream is mainly determined by the message we want to deliver to the observers and the emotions we want to evoke in them. However, in interactive applications, these may vary from time to time and scene to scene with the dynamic change of the virtual environment. Traditionally in CGI, the 3D model of the scene including object geometry, surface properties and lighting is created first, and then the rendered images are enhanced in post-production. To change the stylization of an existing scene, the post-production step is far more efficient in terms of man-hours, since the appearance of every object can be easily altered at once. In video games, this can be done using post-processing effects, which are in general very easy to apply but have the important requirements of real-time performance and temporal coherence, greatly limiting the number of feasible methods. In this work, we were interested in the possibilities that post-processing non-photorealistic rendering (NPR) effects could provide and gathered a collection of methods that can change the overall looking of any game while offering seamless integration. We believe that this kind of post-production can be beneficial for game developers, both for rapid prototyping, in redesigning the appearance of existing games, and in developing games where the visual style changes dynamically with the story.

As a proof of concept, we have implemented the effects in a well-known game engine, Unity, and applied them to existing games including our own development and standard Unity demo projects.

2 NPR Effects for Games

2.1 Edge Enhancement

Identification of contours is one of the most important cornerstones of our visual perception. Edges in images, thus, may serve two purposes: they are, by themselves capable of depicting abstracted objects that our brain is able to recognize, and to aid our perception process by enhancing important visual details (e.g. silhouettes or texture lines). There are two main approaches for edge extraction. The first one tries to mimic artists: lines follow object silhouettes and main geometric features, ridges and valleys; the

standard post-processing implementation looks for discontinuities in the depth and normal maps. The second approach follows a perceptual viewpoint: the sensitivity of our eyes to abrupt changes in light intensity, utilizing classic edge detection methods of image processing searching for pixels of high gradient magnitude. Our implementation follows the work of [Winnemöller 2011] based on Difference-of-Gaussians (DoG), which can produce more artistic results than other edge detection operators. While both geometric and image-space methods have their benefits and drawbacks, their combination produces better results [Redmond and Dingliana 2009]. Additionally, an interesting, brush stroke effect is achieved by setting a constant background for non-edge pixels while using the color map as edge color.

2.2 Texture Simplification

Since the real world (and thereby, images produced by realistic rendering) is visually very complex, relevant information content is suppressed by unimportant details. Studies show that automatic *image abstraction methods* involving image simplification steps to reduce the detailedness of textures and tone complexity can improve user performance in e.g. recognition or search tasks [Winnemöller et al. 2006]. These methods rely on edge preserving filtering such as the bilateral filters and luminance quantization, possibly applied adaptively based on the local gradient magnitude. In our framework we implemented the method of [Winnemöller et al. 2006], later improved in [Kyprianidis and Döllner 2008], as we believed this could produce the most cartoon-like styles among other, similar effects.

2.3 Shadow Recoloring

An important utilization of shadows by artists is to increase the contrast between image areas. As an example, the Chiaroscuro technique tries to catch the viewer's eyes in specific parts of the painting by creating violent contrasts between shadowed and lit parts. Impressionists remove very dark colors in their paintings and represent lit objects with pastel colors and smooth contrast variations whereas shadows are represented by bright colors. Completely black shadows or complementary colors are commonly used in comics. These kinds of effects can be implemented by modifying the hue, saturation, or value components of pixels in the HSV color space, as in [Sauvaget and Boyer 2010].

2.4 Depth Sensation with Varying Abstraction Level

While many effects such as fog, depth of field or atmospheric effects are widely used in game engines to create the sensation of depth, artists have developed other techniques for the same purpose. Since most of their message is placed in the foreground and farther objects serve as the background of the main action, the abstraction level increases with the distance from the viewpoint: richness in color and texture [Sauvaget and Boyer 2009] and the strength of contours [Goodwin et al. 2007] fade with the depth. We use color desaturation with a slight modification of [Redmond and Dingliana 2007], a depth-guided variant of [Winnemöller et al. 2006], and decreasing edge thickness of [Winnemöller 2011].

2.5 Color Palette Modification by Example

The color histogram of an image determines its overall mood; changing it requires to define a mapping on the color palette. The use of standard histogram manipulation effects, however, would impose tedious work to the artists. Automatic *color style transfer*, on the other hand, lets the users to specify example image(s) and the color palette of the input image is adjusted to have a similar color histogram as the exemplar. A simple and fast approach was introduced by Reinhard et al. [Reinhard et al. 2001], which adapts the average color and variance of the input to the exemplar. Different color spaces such as $L\alpha\beta$ [Reinhard et al. 2001] or oRGB [Zhao et al. 2009] may be used to get slightly different results.

3 Future Work

We would like to investigate further possibilities of post processing NPR effects, including stylizations of shadow shapes, stroke based rendering such as hatching, line textures, or more complex example-based rendering techniques as well as seamless application of the effects on third-party games by intercepting standard graphic library calls instead of a Unity implementation. We will add the possibility of selective stylization to highlight desired parts of an image corresponding to specific semantics. We are also working on a user study involving both casual and professional players to see whether re-stylization of existing games enhances user experience and replay value, as well as game developers to testify the ease of use and seamless application of the effects on third-party games.

References

GOODWIN, T., VOLLICK, I., AND HERTZMANN, A. 2007. Isophote distance: a shading approach to artistic stroke thickness. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, ACM, New York, NY, USA, NPAR '07, 53–62.

KYPRIANIDIS, J. E., AND DÖLLNER, J. 2008. Image abstraction by structure adaptive filtering. In *Proc. EG UK Theory and Practice of Computer Graphics*, 51–58.

REDMOND, N., AND DINGLIANA, J. 2007. Adaptive Abstraction of 3D Scenes in Real-Time. In *EUROGRAPHICS '07: Proceedings of the Short Paper Programme*, 77–80.

REDMOND, N., AND DINGLIANA, J. 2009. A Hybrid Approach to Real-Time Abstraction. In *Proceedings of Eurographics Ireland, 2009*.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl. 21*, 5 (Sept.), 34–41.

SAUVAGET, C., AND BOYER, V. 2009. Abstraction of photographs: a comics style approach. In *Proceedings of the 3rd international conference on European computing conference*, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, ECC'09, 215–220.

SAUVAGET, C., AND BOYER, V. 2010. Stylization of lighting effects for images. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2010 Sixth International Conference on*, 43–50.

WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-time video abstraction. *ACM Trans. Graph. 25*, 3 (July), 1221–1226.

WINNEMÖLLER, H. 2011. Xdog: advanced image stylization with extended difference-of-gaussians. In *NPAR*, ACM, J. P. Collomosse, P. Asente, and S. N. Spencer, Eds., 147–156.

ZHAO, H., JIN, X., SHEN, J., AND WEI, F. 2009. Real-time photo style transfer. In *CAD/Graphics*, IEEE, 140–145.