SIGGRAPH Technical Sketch, SIGGRAPH 2003, San Diego, California, USA, 27-31 July 2003

Concise User Control for Texture-By-Numbers Cloning, sketches_0469

Stephen Brooks* Marc Cardle† Neil A. Dodgson‡

University of Cambridge

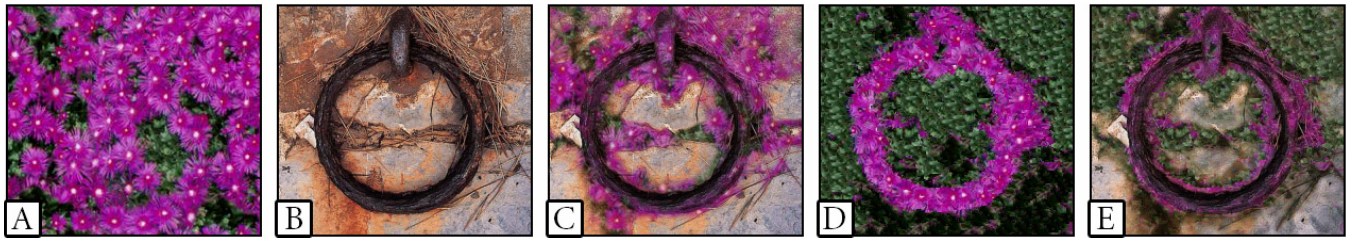

Figure 1 Semantically meaningful texture cloning: A) Texture source for cloning. B) Image target for cloning. C) Cloning with the original source texture. D) Re-arranged version of A using self-similarity masks. E) Cloning of the re-arranged texture.

1 Introduction

Texture mapping is an indispensable tool for achieving realism in computer graphics. Significant progress has been made in recent years with regards to the synthesis and editing of 2D texture images. However, the exploration of user control for semi-automatic texture editing remains an open area of research. We present methods that partially address the semantic and technical limitations of Self-Similarity Based Editing [1]. This is achieved by providing the user with more control over the similarity metric during editing and over "what goes where" during cloning.

2 Improved Texture Editing

In our original paper, Self-Similarity Based Texture Editing, a novel system of concise texture editing was presented which allows the user to make global changes to texture images with minimal user intervention by exploiting the inherent self-similarity of textures [1]. These global operations include painting, cloning and warping. However, as first proposed, the original method has limitations which fall into two categories:

Semantic: For cloning, the tool requires that the cloning image spatially 'matches' the image being cloned into. For example, when Fig.1A is cloned into 1B, the arrangement of the flowers is arbitrary with respect to the ring (Fig.1C).

Technical: For both cloning and painting, the tool does not work as well for textures that contain a high degree of randomness or sharp features. This is due to the smoothing tendency of Gaussian-pyramid neighborhood metrics.

Firstly, we address the semantic limitation by allowing the user to re-arrange the cloning texture (Fig.1A) so that it better matches the target image which it is being cloned onto (Fig.1B), resulting in a more appropriate cloning texture (Fig.1D). We do this by semi-automatically constructing Texture-By-Numbers [2] masks of both the cloning texture (Fig.1A) and the image being cloned onto (Fig.1B). These masks are then used for an Image Analogy [2] guided re-synthesis prior to cloning. For example, we specify that the flowers in Fig.1A, which are labeled with dark purple in the top-left mask of Fig.2, are to be synthesized into the dark purple ring, shown in the top-right mask of Fig.2.

We could force the user to manually construct the Texture-by-Numbers masks by hand. However, this would not be in keeping with the concise nature of Self-Similarity Editing. To automate time consuming Texture-by-Numbers mask constructions, we have developed a new variant of the Self-Similarity toolset that separates an input texture into distinct regions. This tool can be seen as sophisticated "Magic Wand".

The "Wand" compares the multi-scale neighborhoods of all pixels to that pixel the user has selected. Points that have 'Low' similarity to the select point are given one color and those with 'Higher' similarity are assigned another. In fact, the tool can separate the image into an arbitrary number of color sets with the addition of more similarity thresholds along the distance slider. And, for better synthesis, small regions are discarded.

This enhanced "Wand" also allows the user to select multiple similarity points within the texture which together comprise a Boolean similarity expression. In this way, the user can specify that pixels must be like pixel A or pixel B but not pixel C. Once these masks are created, the cloning texture (Fig.1A) is re-arranged (Fig.1D) and subsequently cloned into the target image (Fig.1B), resulting in a more meaningful cloning (Fig.1E).

To address the technical limitations we also explore the use of a Wavelet based similarity metric. Moreover, we give the user even finer control by providing a slider that specifies what proportion of neighborhood versus wavelet responses are to be used in the similarity calculation. By placing more emphasis on wavelet responses the user can thereby cause the self-similarity tool to react more strongly to sharp features in the texture during editing and avoid the problem of excessive smoothing.

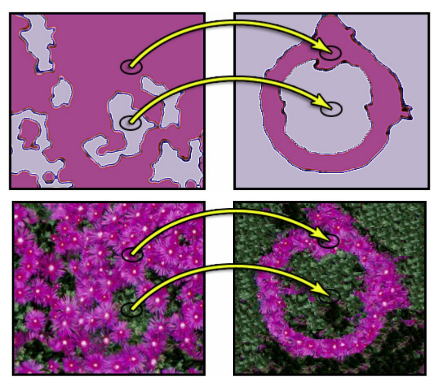

Figure 2 Texture re-arrangement prior to cloning.

References

[1] BROOKS, S., AND DODGSON, N. 2002. Self-Similarity Based Texture Editing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002)*, 21(3), p. 653-656.

[2] HERTZMANN, A., JACOBS, C., OLIVER, N., CURLESS, B., AND SALESIN, D. 2001. Image Analogies. *Proceedings of the 28th Annual Conference on Computer graphics and Interactive Techniques*, p.327-340.

e-mail: {*sb329 | †mpc33 | ‡nad}@cl.cam.ac.uk