

Computational Vision: Example-Based Texture Synthesis

Matthew McMullan

December 11, 2012

1 Overview

Our project focused on the creation of large regions of texture from small example textures. We accomplish this using a method similar to the one described in Parallel Controllable Texture Synthesis, computing our results using a GPU implementation. The general idea is to scale the image up (without interpolation), apply jitter (randomly shift regions of the texture in the image), and to correct the jittered texture by comparing to neighborhoods of the exemplar.

2 Algorithm

As described in [1], there are three main steps to our implementation: upsample, jitter, and correction. Upsampling takes the example texture and creates a larger version, with the resulting texture storing coordinates to values in the exemplar. This texture of coordinates is then used to upsample further in later iterations. We then apply jitter, using simplex noise. The generated noise results in a deterministic texture. This introduces some randomness to reduce the tiled appearance of the texture.

3 Implementation

4 Results

4.1 ‘Good’ Cases

4.2 ‘Bad’ Cases

5 Conclusion

References

- [1] Sylvain Lefebvre and Hugues Hoppe. Parallel controllable texture synthesis. *ACM Trans. Graph.*, 24(3):777–786, July 2005.