A Seamless Texture Color Adjustment Method for Large-Scale Terrain Reconstruction

Hye sun Kim
Electronics and Telecommunications Research Institute, Korea
hsukim@etri.re.kr

Yun ji Ban
Electronics and Telecommunications Research Institute, Korea
banyj@etri.re.kr

Chang joon Park
Electronics and Telecommunications Research Institute, Korea
chjpark@etri.re.kr

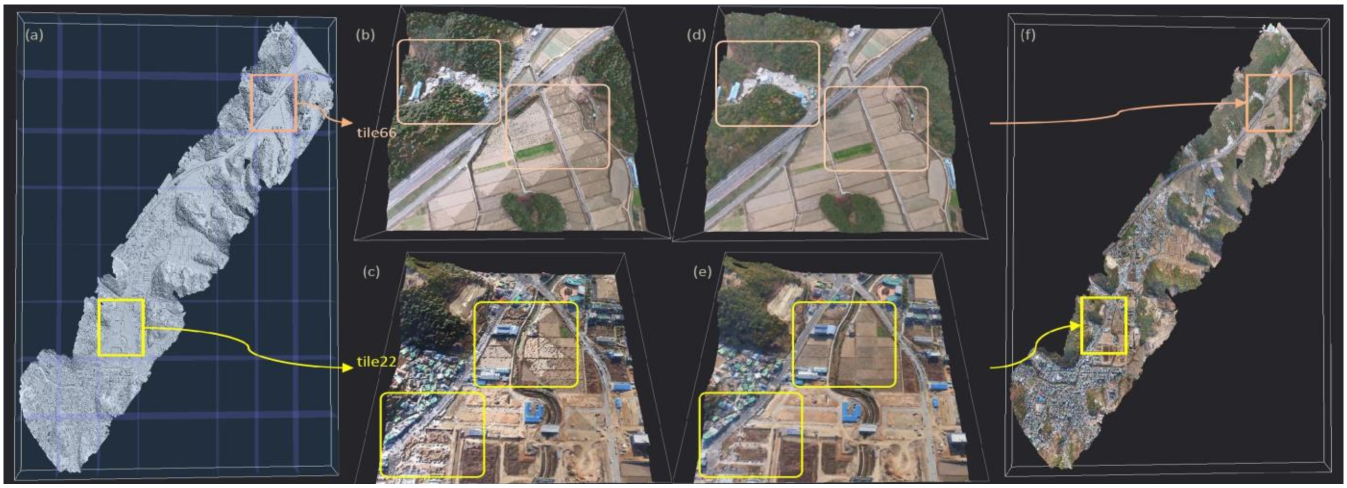

Figure 1: Large-scale terrain texture reconstruction (a) separating large-scale terrain into 8x8 tiles, (b) and (c) before texture color adjustment: texture seams, (d) and (e) after texture color adjustment (f) recombining all tiles

ABSTRACT

We present a technique to generate realistic high quality texture with no seams suitable to reconstruct large-scale 3D terrains. We focused on adjusting color difference caused by camera variations and illumination transition for texture reconstruction pipelines. Seams between separated processing areas should also be considered important in large terrain models. The proposed technique corrects these problems by normalizing texture colors and interpolating texture adjustment colors.

CCS CONCEPTS

• **Computing methodologies → Computational photography**

KEYWORDS

Texture reconstruction, Automatic color adjustment and 3D terrain reconstruction

SIGGRAPH '18 Posters, August 12-16, 2018, Vancouver, BC, Canada
ACM 978-1-4503-5817-0/18/08.
10.1145/3230744.3230788

ACM Reference format:
Hye sun Kim, Yun ji Ban, and Chang joon Park. 2018. A Seamless Texture Color Adjustment Method for Large-Scale Terrain Reconstruction. In *Proceedings of SIGGRAPH '18 Posters, Vancouver, BC, Canada, August 12-16, 2018,* 2 pages.
DOI: 10.1145/3230744.3230788

1 INTRODUCTION

Various techniques have been proposed for image-based 3D reconstruction, incorporating several complex pipelines; including camera pose estimation and 3D geometric data, 3D mesh, and texture reconstruction. The last process, texture reconstruction projects and maps the best-viewed image onto each face of the mesh, producing an impressive and realistic 3D mesh. However, problems can arise by matching one image per face. Texture seams is occurred when neighboring faces select different images with discontinuous colors, as shown Fig. 1 (b). Although some studies have addressed this problem by blending multiple images for one face, we excluded blending because this tends to degrade texture quality by omitting low frequency details [Bernardini et al. 2001; Weachter et al. 2014]. Weachter et al. [2014] proposed how to select the best view image for each faces among visible images and how to optimize by mosaicking. However, they correct texture seams locally without considering reconstructing objects become very large.

To reconstruct large-scale terrains at once is very inefficient, because there might be a shortage of computing resources and handling and rendering terrains might be difficult later. Generally, the way to split a huge terrain into smaller tiles is adopted, as shown in Fig. 1 (a). Images and 3D data for each tiles are reconstructed to mesh and texture separately. However, this process raises boundary continuity problems between tiles, and it is essential to smooth textures between tiles globally, even though each tiles are processed locally.

This paper proposes a technique to smooth texture seams with high quality, simultaneously correcting discontinuities caused by image color difference and tile-based processing.

2 TEXTURE COLOR ADJUSTMENT

Fig. 2 shows internal data images used to reconstruct texture for the indicated tile Fig. 1 (b) and (d). Fig. 2 (b) represents selection of the best view image in its unique color, where a collection of faces with the same color is defined as a texture patch. Texture seams appear at the boundary between texture patches, comparing Fig. 1 (b) to Fig. 2 (b), these have to be corrected.

Texture color adjustment can be achieved by controlling vertex colors of each faces added to the seamed texture, then the inside of the face is corrected by GPU shading. How to determine the vertex color is consists of the following two steps.

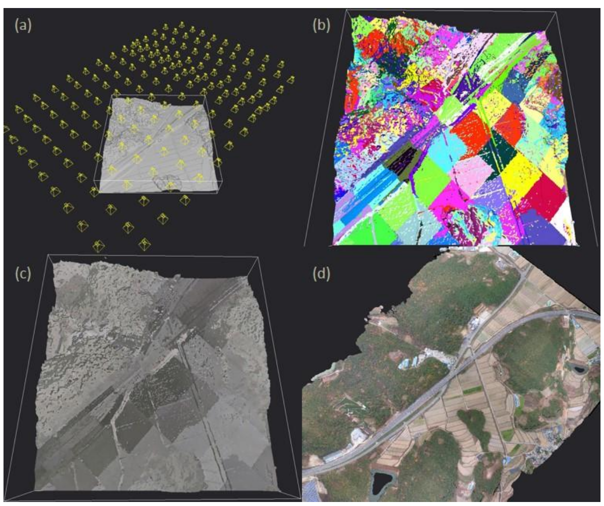

Figure 2: (a) input tile mesh and camera setup, (b) texture patch: the face set of same selection of the image, (c) color adjustment map, (d) result terrain.

2.1 Texture color normalization: boundary vertex

Boundary vertices are located at the boundary between texture patches or separated tiles, where texture seams occur. For example, orange color points on Fig. 3 (a) are boundary vertices. To adjust the boundary vertex colors, we first determine the target color by averaging candidate images' colors, represented on the first term of equation 1. Candidates mean images that the vertex is visible as shown Fig. 3 (c). Since all tiles and vertices share images and have common candidates from same positions, target colors could be same globally. The adjustment color is the substance value by the current selected image's color.

In equation 1, VC_k is the adjustment color of k_{th} vertex, and $IC^s_{x,y}$ is the pixel (x, y) color on a best view image s, $IC^i_{x,y}$ is a pixel color of i_{th} candidates. The n means sampling window size on image.

$$VC_k = \frac{\sum_i^{i\in candidates}\sum_{x=px_i,y=py_i}^{n,n} IC^i_{x,y}}{|candidates|} - \sum_{x=px_s,y=py_x}^{n,n} IC^s_{x,y} \qquad (1)$$

2.1 Texture color interpolation: inner vertex

Once the adjustment colors of boundary vertices are determined, inner vertex colors should be corrected smoothly (Fig 3, blue color points). Since the texture patch is a closed planar graph without self-intesection, we use the Mean Value Coordinates (MVC) technique to interpolate adjustment colors of inner vertices [Hormann et al. 2006]. Fig. 3 (b) shows a result of the vertex adjustment color map.

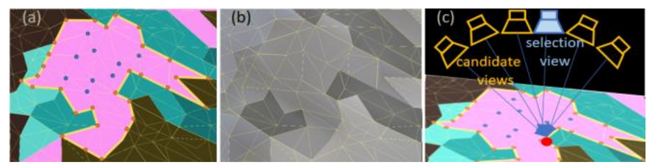

Figure 3: (a) boundary (orange) and inner (blue) vertices of an example texture patch (pink), (b) color adjustment map, (c) candidate and selection view

3 CONCLUSIONS

This paper proposed a technique for high quality texture color adjustment to reconstruct large-scale terrain, thereby removing any texture discontinuity between texture patches and tiles. The proposed technique is very simple and efficient, requiring only few seconds to correct texture colors for large scale images: 5 seconds for 1,000,000 face mesh and 161 images (4K) using Intel ® Core™ i7-3770K CPU 3.5GHz, 16 GB RAM, NVidia GeForce GTX 1080 and Windows 10 OS.

ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2016-0-00008), Synthetic Battlefield Generation and Visualization from Aerial Imagery Captured at Multiple Altitudes)

REFERENCES

Michael Waechter, Nils Moehrle and Michael Goesele. 2014. Let There Be Color! Large-Scale Texturing of 3D Reconstructions. ECCV 2014, pp 836-850

Faust Bernardini, Ioana. M. Martin and Holly Rushmeier. 2001. High-Quality Texture Reconstruction from Multiple Scans. IEEE Transactions on Visualization and Computer Graphics, Volume 7, Issue 4, pages 318-332

Kai Hormann and Michael. S. Floater. 2006. Mean Value Coordinates for arbitrary planar polygons, Journal ACM Transactions on Graphics(TOG), Volume 25, Issue 4, pages 1424-1441