3D Reconstruction using Spherical Images

P.D. Williams[1], A. Hilton

[1] Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK; pw00001@surrey.ac.uk

Keywords: Spherical Stereo, Texturing, 3D Reconstruction

Abstract

A process is presented for the creation of immersive high-dynamic range 3D models from a pair of spherical images. The concept of applying a spherical Delauney triangulation to 3D model creation is introduced and a complete texturing solution is given.

1 Introduction

Immersive 3D models of real locations have many possible applications. Photo realistic models would be particularly useful as virtual sets for film and game production. This Paper describes a process that is being developed that makes use of a pair of high-dynamic range spherical images to create an immersive 3D model from only a pair of images.

The only paper to date covering spherical stereo is by Shigang Li and Kiyotaka Fukumori[1]. In which the epipolar equation is derived for spherical images and a data structure is presented for the representation of spherical images. The sample results in [1] seem to display errors in their texturing method. This paper describes a complete process including a robust texturing technique and what has been found to be the most appropriate triangulation method.

2 Process Overview

Feature Detection and Correspondence: The first stage in the process is to detect suitable features in one image for which to find correspondaces in the other image. The current implementation uses the SUSAN corner detector[3] for this. The corresponding epipolar line in the second image is then searched for the best match with the corner using normalised cross correlation on patches of pixels. The best values that are above a threshold are then taken as valid correspondences and their 3D position reconstructed.

Triangulation: For display as a 3D model these points need to be converted into a triangle mesh. The most suitable triangulation for fully surrounding 3D points was found to be one formed on the surface of a sphere, as it covers all directions whilst ensuring all triangles are completely visible from the models origin. The spherical Delauney triangulation provided in the FORTRAN library STRIPACK[2] has been found to be a good implementation for this. In order to compensate for errors in the correspondence stage a smoothness constraint is applied to the triangle mesh, removing nodes that cause large angles in the mesh.

Texturing: To texture the triangle mesh a cube environment map image format is used. For each 3D point the corresponding cube face is selected and homogeneous texture coordinates are calculated compensating for foreshortening. Using a cube environment map requires triangles that cross more then one cube face to be clipped along the cube edges. An algorithm has been developed to solve this for all possible cases.

Sample virtual viewpoints obtained from the process are shown in Figure 1.

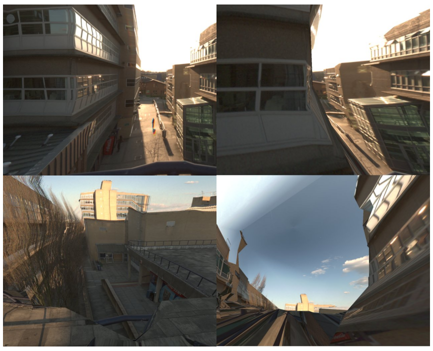

Figure 1: Sample Virtual Views

3 Conclusions

A complete process has been developed. The texturing stage is considered to be complete but further work is required on the earlier stages to improve the accuracy of the 3D reconstruction. The main sources of distortion are caused by insufficient features being detected at the start, errors in corresponding the features between the two images, and the uninformed positioning of triangle arcs. It is expected that work currently in progress, using different feature detectors and re-sampling the input image to give even pixel sizes, will improve the models obtained. Greater improvements should come from the use of edge detection to better place the triangle arcs.

References

[1] Shigang Li and Kiyotaka Fukumori. Spherical stereo for the construction of immersive vr environment. In *Proceedings of the IEEE Virtual Reality*, pages 217–222, 2005.

[2] Robert J. Renka. Algorithm 772. STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere. 23(3):416–434, September 1997.

[3] S.M. Smith. A new class of corner finder. In *Proc. 3rd British Machine Vision Conference*, pages 139–148, 1992.