PanoBasic: Toolbox for panoramic image processing

Yinda Zhang

July 16, 2014

1 Introduction

Panorama image can be considered as a 2D display of a texture map on a 3D sphere via Equirectangular projection (think about a world map). Due to the projection, the panorama is warped, e.g. straight lines become curves, and texture around polar is severely distorted, so that many previous methods cannot be applied on it directly. This toolbox provides supporting functions for panoramic image processing. It contains supports for:

a. Manipulate image on 3D sphere.

b. Projection between panorama and normal perspective image.

c. Basic image processing, e.g. line segment detection, color image segmentation, etc..

d. Basic 3D reconstruction, e.g. how to get 3D information base on geometric assumption.

If use this code as part of your project, please cite:

[3] Yinda Zhang, Shuran Song, Ping Tan, Jianxiong Xiao,
PanoContext: A Whole-room 3D Context Model for Panoramic Scene Understanding, Proceedings of the 13th European Conference on Computer Vision (ECCV2014)

2 Functions

To get familiar about the toolbox, it is strongly recommended to run the **demo.m**. It runs most of the major functions on an example panorama image.

2.1 Coordinate Transform

A panorama is a texture map on a sphere. Every pixel on image corresponds to a point on a unit sphere, as well as a viewing direction in 3D space. So a pixel can be represented by its image location on panorama (X, Y), 3D direction (U, V) (azimuth, elevation), or position on unit sphere (x, y, z). The transferring functions among three representations are included in folder **CoordsTransfrom**.

2.2 Projection between Perspective Views and Panorama

A panorama can be projected to a serial of normal perspective images. Each perspective view is defined by a 3D view direction d, a focal length R, and the image size S. The corresponding image plane then is a 3D plane cutting a ball with radius as R at point $R * d$ on the sphere. By shooting rays from ball center passing through the ball sphere and the cutting plane, the correspondence between the panorama and the perspective view is built. Relevant functions can be found in folder **Projection**.

2.3 Uniformly Sampling Directions

To uniformly get information from panorama, we need to uniformly sample directions in 3D space. Uniformly sampling pixels on panorama is not correct as it will severely bias to polar region. Call function in folder **icosahedron2sphere**[1] to get uniformly sampled vectors. Since the computation may be slow when sampled densely, we also provide some pre-computed results.

2.4 Line Segment Detection

Straight lines in 3D space or 2D perspective images become curves on panorama. To leverage previous line segment detection work, we project panorama to overlapping perspective views and run LSD[2] on each view. Then line segments from all views are combined to generate line segment map for panorama. Relevant functions can be found in folder **BasicProcessing/VpEstimation**.

2.5 Vanishing Point Estimation

Under Manhattan World Assumption, the three orthogonal vanishing directions can be obtained by Hough Transform on 3D sphere. Please refer to our paper[3] for more details. Relevant functions can be found in folder **BasicProcessing/VpEstimation**.

2.6 Image Segmentation

We modify graph-based image segmentation(GBS)[4] for panorama. In original GBS, the graph is built as pixel grid, which is not suitable for panorama. For example, on panorama, all pixel on the first row are actually corresponding to one point in 3D space; also the left column is actually connected with the right column. To handle panorama geometry, we uniformly sample directions on sphere as vertices and build graph on them. The relevant functions can be found in folder **BasicProcessing/segmentation**.

2.7 Image Rotation

The panorama can be rotated to a new coordinates system defined by three orthogonal directions. It should be done in 3D space. Functions can be found in folder **BasicProcessing/rotation**.

2.8 Polygon-defined Region

A sequence of points in clockwise or counter-clockwise order can define a region on panorama. In 3D space, connecting these points with camera center forms a cone. We provide function to test whether a given direction is inside/outside a cone. This function is especially helpful when computing feature from a particular region. Functions can be found in folder **BasicProcessing/polygonRegion**.

2.9 Basic 3D Reconstruction

Under some assumption, some simple 3D reconstructions can be done by given a single panorama. For example, if we assume the ground plane is horizontal, and camera height is 160mm, the exact 3D position of a point on ground plane can be computed according to its 2D location on panorama. To do so, we only need to computer the intersection of the ground plane and a 3D ray passing through the point on sphere and camera center. Such toy example is provided in **demo.m**.

References

[1] Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L.: Image-based street-side city modeling. Proceedings of ACM SIGGRAPH Asia (2009)

[2] von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a Line Segment Detector. Image Processing On Line (2012)

[3] Zhang, Y., Song, S., Tan, P., Xiao, J.: PanoContext: A whole-room 3D context model for panoramic scene understanding. In: ECCV. (2014)

[4] Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)