Upright Adjustment of 360 Spherical Panoramas

Jinwoong Jung[*] POSTECH, Korea
Joon-Young Lee[†] Adobe Research, USA
Byungmoon Kim[‡] Adobe Resaerch, USA
Seungyong Lee[§] POSTECH, Korea

Figure 1: Our method automatically uprights a single 360 spherical panorama image (left) by aligning the perceived upward direction of the scene to the vertical axis of the panorama image, producing the result image (right) that is much more pleasant to look at. To achieve this, we detect edges, find great circles, analyze vanishing points, and finally solve for optimal 3D rotation.

ABSTRACT

With the recent advent of 360 cameras, spherical panorama images are becoming more popular and widely available. In a spherical panorama, alignment of the scene orientation to the image axes is important for providing comfortable and pleasant viewing experiences using VR headsets and traditional displays. This paper presents an automatic framework for upright adjustment of 360 spherical panorama images without any prior information, such as depths and Gyro sensor data. We take the Atlanta world assumption and use the horizontal and vertical lines in the scene to formulate a cost function for upright adjustment. Our method produces visually pleasing results for a variety of real-world spherical panoramas in less than a second.

Keywords: spherical panorama, upright adjustment

Index Terms: I.4.3 [Image Processing and Computer Vision]: Enhancement—Registration

1 INTRODUCTION

Spherical panoramas (360 panoramas) are images with 360 degrees of horizontal and 180 degrees of vertical field of views. In the past, 360 panoramas used to be niche, typically created by skilled professionals. Recently this has been changing as mobile panorama applications have been released, and the use of 360 panoramas is accelerated as new 360 cameras become available.

In typical 360 panorama images taken with a low-cost camera, e.g., Ricoh Theta S ($350), wavy horizons and slanted objects due to slight camera tilts and rolls are common, as shown in Figure 1. By design, 360 cameras do not provide stable grips or view finders while users often take lots of photos with hand-held cameras. The correction of this mis-orientation will greatly improve the experience as the user feels standing straight in the scene. This kind of correction is called *upright adjustment* [2].

In this paper, we present an automatic method for upright adjustment of 360 spherical panorama images. We take the Atlanta world assumption [3] for the scene, and find horizontal and vertical lines in the input panorama, which are then mapped onto great circles on the sphere used for representing a spherical panorama image. By imposing the desired properties on the great circles from vertical lines and the vanishing points from horizontal lines, we formulate a cost function for upright adjustment. Then, the optimal rotation to resolve the mis-orientation of the camera can be obtaining by minimizing the cost function, where a rotation is represented as an update of the north pole position of the sphere.

The main contributions of our work are summarized as follows:

- Analysis of the upright adjustment problem of spherical panoramas, providing a simple but effective formulation with a cost function
- Fast method for optimizing the cost function to find an optimal rotation that resolves the mis-orientation of the camera

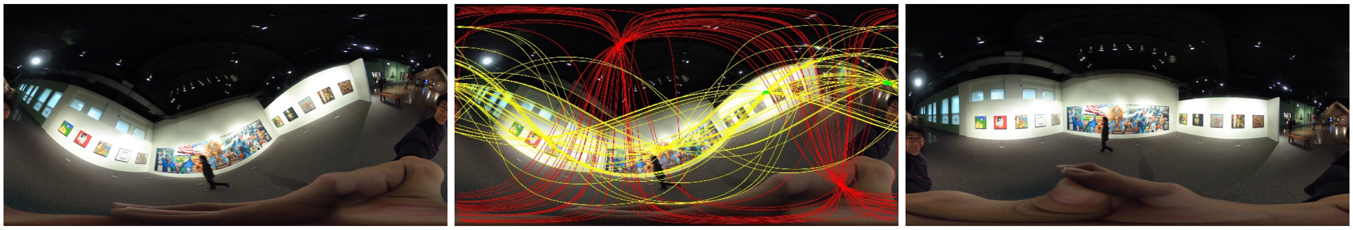
2 UPRIGHT ADJUSTMENT PROCESS

The rotation needed for upright adjustment of a spherical panorama has 2 DOF, i.e., *tilt* and *roll*, as *pan* is not related to undesirable inclination of the camera. We represent our desired 2 DOF rotation as the update of the north pole position on the sphere, where the position is specified by a unit vector.

Line segment detection We use LSD algorithm [4] to find line segments from the input spherical panorama image, although any line detection algorithm, such as EDLines [1], can also be used. The detected line segments are usually short, because of the circular distortions of lines in a spherical panorama. We consider a detected line segment as a part of the projection of a 3D line onto the sphere. Parts of a 3D vertical line are mapped onto vertical line segments in a spherical panorama except around the north and south poles when there is no tilt and roll of the camera. In the case of a 3D horizontal line, regardless of the line direction, the projected line segments are

*e-mail: ca00229@postech.ac.kr
†e-mail: jolee@adobe.com
‡e-mail: bmkim@adobe.com
§e-mail: leesy@postech.ac.kr

2017 IEEE Virtual Reality (VR)
March 18-22, 2017, Los Angeles, CA, USA
978-1-5090-6647-6/17/$31.00



(a) (b) (c)

Figure 2: Overall process. To find the optimal rotation for upright adjustment, we first extract line segments, then obtain vertical great circles and horizontal vanishing points by mapping the line segments into a spherical Hough space, and finally optimize a cost function. (a) input image. (b) great circles. Green dots are vanishing points computed from horizontal (yellow) lines. (c) upright adjustment result.

almost horizontal in a spherical panorama when they are far away from the vanishing points.

In a given spherical panorama captured with a mis-oriented camera, the vertical and horizontal line segments from 3D vertical and horizontal lines would appear slanted reflecting the camera orientation. We define the *slanted angle* of a line segment as the angle from the x-axis of the input image. If the slanted angle of a detected line segment is less than γ_1, it is classified as horizontal. If the angle is greater than γ_2, the line segment is classified as vertical. Otherwise, the line segment is discarded. We use $\gamma_1 = \frac{\pi}{6}$ and $\gamma_2 = \frac{\pi}{3}$ for all examples in our experiments.

Great circle detection To recover the great circles corresponding to 3D vertical and horizontal lines in the scene, we apply spherical Hough transform to the detected horizontal and vertical line segments. For a line segment, the two end points are points on the sphere and determines a great circle. We quantize the unit hemisphere into the resolution of $m \times n$, and accumulate the unit vectors of the great circles produced by horizontal and vertical line segments, where the lengths of line segments are used for the weights in the accumulation. In our experiments, we use $m = 360$ and $n = 90$. After accumulating all line segments, we select top t_1% horizontal and top t_2% vertical great circles in terms of accumulated weights, where the maximum numbers of the selected great circles are limited by k_1 and k_2, respectively. We use $t_1 = t_2 = 10$ and $k_1 = k_2 = 50$. Figure 2(b) visualizes the detected great circles overlaid on the input image.

Vanishing point detection Using the great circles recovered from horizontal line segments, we find vanishing points. We again quantize the unit hemisphere into the resolution of $m \times n$, and rasterize the great circles to find intersections. The weight of a great circle accumulated in the previous spherical Hough transform is added to the quantized cells in the rasterization process. We finally choose top t_3% non-zero cells containing the largest weights to determine vanishing points from parallel horizontal lines, where k_3 is the maximum allowed number of chosen cells. We use $t_3 = 10$ and $k_3 = 30$. Figure 2(b) shows detected vanishing points that are the intersections of great circles from horizontal lines.

Formulation We formulate the cost function for upright adjustment of a spherical panorama as follows.

$$E(\overrightarrow{P}) = \alpha \sum_i (\overrightarrow{v_i} \cdot \overrightarrow{P})^2 + \beta \sum_j (\overrightarrow{h_j} \cdot \overrightarrow{P})^2 + \lambda (1 - \overrightarrow{y} \cdot \overrightarrow{P})^2, \quad (1)$$

where $\overrightarrow{v_i}$ is the unit vector perpendicular to a great circle obtained from vertical lines and $\overrightarrow{h_j}$ is the unit vector corresponding to a vanishing point computed from horizontal lines. The last term is for regularization to prevent a drastic change of the north pole from the original position (y-axis). α, β, and λ are user-specified relative weights of the terms. By finding a unit vector $\overrightarrow{P}$ that minimizes Eq. (1), we can obtain the updated position of the north pole that resolves the tilt and roll of the camera applied when the input spherical panorama was being taken. We used $\alpha = 1.0$, $\beta = 0.5$, and $\lambda = 10.0$ for all experiments in the paper.

Iterative adjustments After we have resampled the input spherical panorama image using the rotation computed for upright adjustment, vertical line segments would become more straightened up with reduced circular distortions, especially near the horizon. To obtain a more accurate upright adjustment result, we repeat the optimization process using Eq. (1), where the output of the current iteration becomes the input of the next iteration. As the iteration goes, the great circles from vertical line segments become more accurate, and the north pole can be determined more accurately as the intersection of the great circles by minimizing Eq. (1). Figures 1 and 2(c) show the final upright adjustment results.

3 Results

We implemented our method using C++ and the OpenCV library on a 64-bit Windows PC with an Intel i7-6700K 4.00GHz CPU and 32GB RAM. For an input image of size 5376 × 2688, it takes a few hundred milliseconds (less than one second) to obtain the optimal north pole position for upright adjustment. The supplementary material includes additional examples of our upright adjustment of spherical panoramas, demonstrating the effectiveness of our method. As our method works fast even with our unoptimized implementation, it could be used as an on-demand pre-processing step to calibrate the initial orientations of spherical panorama images in various 360 VR applications, such as 360 image viewing and editing, and 360 video stabilization.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant (R0126-16-1078), the National Research Foundation of Korea (NRF) grant (NRF-2014R1A2A1A11052779), and Korea Creative Content Agency (KOCCA) grant (APP0120150512002), funded by the Korea government (MSIP, MCST).

References

[1] C. Akinlar and C. Topal. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633–1642, 2011.

[2] H. Lee, E. Shechtman, J. Wang, and S. Lee. Automatic upright adjustment of photographs with robust camera calibration. *IEEE PAMI*, 36(5):833–844, 2014.

[3] G. Schindler and F. Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proc. CVPR*, pp. 203–209, 2004.

[4] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE PAMI*, 32(4):722–732, 2010.