

A GREAT CIRCLE ARC DETECTOR IN EQUIRECTANGULAR IMAGES

Seon Ho Oh and Soon Ki Jung

*School of Computer Science and Engineering, College of IT Engineering, Kyungpook National University,
80 Daehak-ro, Buk-gu, Daegu 702-701, Republic of Korea*

Keywords: Line Segment Detection, Equirectangular Images, Great Circle Arc Detection, Number of False Alarms.

Abstract: We propose a great circle arc detector in a scene represented by an equirectangular image, i.e. a spherical image of the 360° longitude and 180° latitude field of view. Since the straight lines appear curved in equirectangular images, the standard line detection algorithm cannot be used directly in this context. We extend the LSD method (Gioi et al., 2010) to deal with the equirectangular images instead of planar images. So the proposed method has most of the advantages of the LSD method, which gives accurate results with a controlled number of false detections but requires no parameter tuning. This algorithm is tested and compared to other algorithm on a wide set of images.

1 INTRODUCTION

Spherical panoramas are increasingly popular so that the web services such as Flickr enable sharing of this kind of photos. The development of computational photography techniques such as stitching and composing makes it easy for people to create the spherical panoramas. Furthermore, many map services such as Google Street View provide a 360-degree street-level view which allows us to explore places around the world. Accordingly, many conventional image processing techniques will be adopted into the panoramic scene.

Line segment detection is a classical problem in computer vision. A typical method first applies Canny's edge detector (Canny, 1986) followed by the Hough transform (Ballard, 1981) for extracting all lines that contain a number of edge points exceeding a threshold. These lines are thereafter cut into line segments by using gap and length thresholds. However, the line detection methods based on Hough transform are suffering from some problems. First, a high density of edge points such as textured zones gives many false detections. Second, it is hard to find a good threshold. To overcome these problems, various researches were conducted. Burns *et al.* introduced a method to use only gradient orientations (Burns et al., 1986). Desolneux *et al.* proposed a method to control the number of false positives (Desolneux et al., 2000). And then, von Gioi *et al.* proposed Line Segment De-

tor (LSD), which is an improved method that combines the methods of Burns *et al.* and Desolneux *et al.* (Gioi et al., 2010). However, it cannot be directly applied to panorama case in which straight lines of the scene usually appear curved.

Line segment detections for the panoramic images are proposed by several researchers. Fiala and Basu introduced the panoramic Hough transform on the parameter space in 2D for a panoramic-catadioptric image (Fiala and Basu, 2002). They mapped the edge pixels to panoramic Hough parameter space and then extracted lines same as standard Hough transform. Sacht *et al.* also proposed a method to detect lines in equirectangular images (Sacht et al., 2010). The key idea for this method is to use perspective projection that preserves the line features, and Hough transform with local analysis. They project the viewing sphere onto the unit cube and apply the bilateral filter. Thereafter, they perform local analysis proposed by Szenberg (Szenberg, 2001) to avoid the problems of Hough transform and discard the cells where no predominant direction. However, both methods are based on the Hough transform. Therefore, they cannot be free from the problems of Hough transform as mentioned before.

The aim of this paper is to present an effective algorithm that cumulates most of the advantages of the previous line detection algorithm on the planar image and extends it to equirectangular images. We choose LSD proposed by von Gioi *et al.* because the method

made a breakthrough in the extraction of the line segments. To do this, we first introduce a new representation of a great circle arc which means a line segment on the equidistance image. With this representation, we determine the line-support region and associate it with a line segment.

The remainder of the paper is organized as follows. In Section 2, we introduce the geometry of line segments projected on equirectangular images. Section 3 describes how to extend the LSD method for the equirectangular images. Some experimental results are illustrated in Section 4. Finally, we describe the conclusion and future work in Section 5.

2 GREAT CIRCLE ARCS IN EQUIRECTANGULAR IMAGES

In the photography, a panorama is a broad term for an image with elongated field of view. Its format is defined by the projection of a 3D spherical globe into a 2D image. Two main formats are widely used; equirectangular and cubic, but we focus on the equirectangular images in the paper.

Prior to explain the equirectangular projection, we briefly describe the cylindrical equidistant projection, whose transformation equations are

$$\begin{aligned} x &= (\lambda - \lambda_0) \cos \phi_1, \\ y &= \phi, \end{aligned} \quad (1)$$

and the inverse formulas are

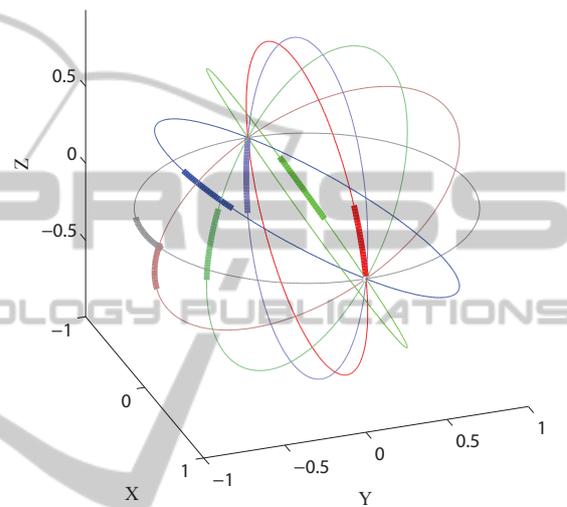
$$\begin{aligned} \phi &= y, \\ \lambda &= \lambda_0 + x \sec \phi_1, \end{aligned} \quad (2)$$

where λ is the longitude, λ_0 is the central meridian of the projection, ϕ is the latitude, and ϕ_1 is the standard parallels (north and south of the equator) where the scale of the projection is true. An equirectangular projection is a cylindrical equidistant projection, in which the horizontal and vertical coordinates are the longitude and the latitude, respectively, so the standard parallel is taken as $\phi_1 = 0$ (Weisstein, 2011).

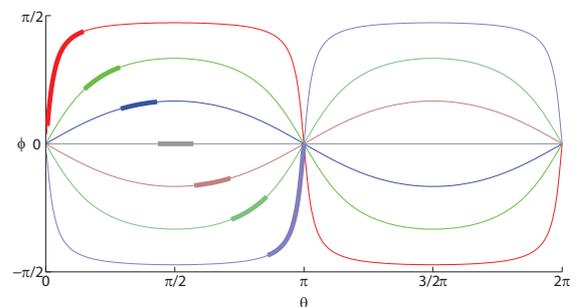
In general, a line segment can be defined as a straight region where many of its points share roughly the same image gradient angle. However, the points on a great circle arc in the equirectangular image do not share the image gradient. Accordingly, we cannot use the image gradient angle directly. Even so, there is a geometric relationship related to the image gradient angle.

In the spherical panoramic imaging, a 3D point is projected into a point on the unit sphere where the

center of projection is the center of the sphere. Similarly, a 3D line segment is projected to its corresponding arc segment on a great circle of the unit sphere. Finally, the arc on a great circle forms a curve segment in the equirectangular image as shown in Fig. 1. Thick arcs on the great circles are given as in Fig. 1 (a), and their corresponding curve segments are illustrated in Fig. 1 (b). As shown in Fig. 1 (a), a great circle is defined by the normal vector for the plane passing through the center of the sphere and the great circle.



(a) The arcs on the great circle of the unit sphere.



(b) The curve segments on the curve in the equirectangular image.

Figure 1: The relationship between the arc on the great circle of the unit sphere and the curve segments in the equirectangular image.

With the relationship described above, it is possible to recover the normal vector for a great circle (NGC) from the pixel and its level-line angle. In addition, the points on the arc share the same NGC. Therefore, it is able to define the arc on the great circle of the unit sphere (which corresponds to the 3D line segment in the scene) with its center, length, and NGC.

This representation is what we call the great circle arc (GCA) and the details will be described in Section 3.3.

3 ALGORITHM

In this section, we explain the algorithm to detect great circle arcs (GCAs) in equirectangular images. First we overview the complete algorithm and then explain the details.

3.1 Overview

Our algorithm extracts GCAs in three steps; region growing, GCA approximation, and validation. Algorithm 1 describes the pseudo code for the proposed algorithm. There are three parameters: ρ , τ , and ϵ . The parameter ρ is a threshold that the points whose gradient magnitudes are smaller than ρ are not considered as seeds of region growing in order to cope with the quantization effect of image intensity values (Gioi et al., 2010; Desolneux et al., 2002). And τ is the angle tolerance used in the search for GCA-support regions. A small value is more restrictive, leading to an over-partition of line segments, and a large value causes unexpected merging of unrelated one. The last parameter ϵ is not a critical one; one can safely set its value to 1 once for all. See (Gioi et al., 2008).

Algorithm 1: GCA detector.

input: An image I , three parameters ρ , τ and ϵ .
output: A list out of lines in equirectangular images.

- 1: (LLAngles, GradMod, OrderedListPixels) \leftarrow Grad(I, ρ)
- 2: Status(allpixels) \leftarrow NotUsed;
- 3: **foreach** pixel p in OrderedListPixels **do**
- 4: **if** Status(p) = NotUsed **then**
- 5: region \leftarrow RegionGrow(p, τ , Status);
- 6: gca \leftarrow GCAApprox(region);
- 7: nfa \leftarrow NFA(gca);
- 8: (nfa, region) \leftarrow ImproveGCA(gca);
- 9: **if** nfa $<$ ϵ **then**
- 10: Add gca to out;
- 11: Status(region) \leftarrow Used;
- 12: **end**
- 13: **end**
- 14: **end**

The subroutine *Grad* computes the image gradient and gives three outputs: the level-line angles, the gradient magnitude, and an ordered list of pixels. The pixels are classified into bins by their gradient magnitudes.

The list starts with the pixels belonging to the bin with the highest gradient magnitude and is roughly ordered with decreasing gradient magnitude. *Status* is used to keep track of pixels used by GCA-support regions. Starting from the first pixels in the list, *RegionGrow* is used to obtain a GCA-support region. Then, *GCAApprox* gives a GCA approximation of the region and *NFA* computes the number of false alarms (NFAs) of the GCAs. The routine *ImproveGCA* tries several perturbations to the initial approximation in order to get a better approximation.

3.2 GCA-support Regions

Each region starts with just one pixel and its NGC is computed with the level-line angle and the (θ, ϕ) coordinate of the pixel. Then, the pixels adjacent to the region are tested; the pixels whose NGCs are equal to the NGC of the region (ngc_{region}) up to a certain precision are added to the region, and their status is marked as Used. For each iteration, the NGC of the region is updated with the mean of NGCs for all pixels in the region. This process is repeated until no new pixel is added. The pseudo code is illustrated in Algorithm 2.

Algorithm 2: Region growing.

input: A starting pixel (x, y) , an angle tolerance τ , and *Status* where pixels used by other regions are marked.
output: A list *Region* of pixels.

- 1: *Region* \leftarrow (x, y) ;
- 2: $ngc_{region} \leftarrow$ ComputeNGC(x, y , LevelLineAngle(x, y)));
- 3: $ngc_{acc} \leftarrow ngc_{region}$
- 4: **foreach** pixel p in *Region* **do**
- 5: **foreach** \bar{p} neighbor of p and Status(\bar{p}) \neq Used **do**
- 6: $ngc_{\bar{p}} \leftarrow$ ComputeNGC($\bar{p}.x, \bar{p}.y$, LLAngle($\bar{p}.x, \bar{p}.y$)));
- 7: **if** $ngc_{\bar{p}} \cdot ngc_{region} > \cos(\tau)$ **then**
- 8: Add \bar{p} to *Region*;
- 9: Status(\bar{p}) \leftarrow Used;
- 10: $ngc_{acc} \leftarrow ngc_{acc} + ngc_{\bar{p}}$;
- 11: $ngc_{region} \leftarrow$ normalize(ngc_{acc});
- 12: **end**
- 13: **end**
- 14: **end**

The subroutine *ComputeNGC* computes the normal of the great circle passing through the given pixel, whose tangential vector is its level-line angle.

In LSD (Gioi et al., 2010), τ is the angle tolerance used in the search for line-support regions. And it is also used to set the first angle tolerance used in the

NFA computation. However, we do not use level-line angle to determine GCA-support region directly. So we define a new criterion to determine GCA-support region. Assume that the GCA-support region is given and its projection to a unit sphere is shown in Fig. 2. Then, the NGCs of the points in the region form a gray cone whose opening angle is τ . In this manner, we define $\cos(\tau)$ as a new criterion to determine GCA-support region. So that all NGCs of points in the GCA-support region are laid in the cone. In other words, only the pixels whose the inner product with ngc_{region} is greater than $\cos(\tau)$ are considered.

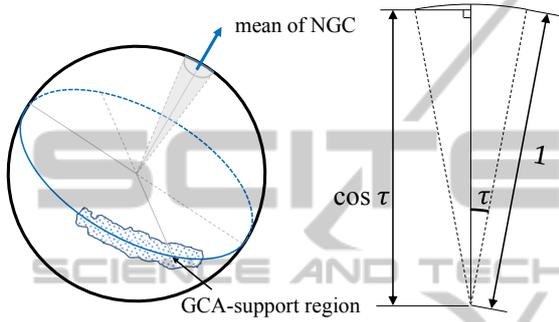


Figure 2: Region growing criteria defined by angle tolerance τ .

3.3 GCA Approximation of Regions

Prior to the validation step, it is necessary to simplify the GCA-support region (a set of pixels) to a GCA (a geometrical object). In general, a line support region in a planar image can be represented by a rectangle. However, a GCA-support region in an equirectangular image is bounded by a curved region. So we define a GCA with its center, length, width, and NGC as shown in Fig. 3.

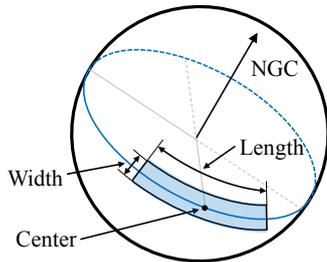


Figure 3: GCA is characterized by a region determined by its center point, NGC, length and width.

In order to obtain a GCA, the center of the GCA-support region should be determined first. In our algorithm, the center of mass is used to compute the center of the GCA-support region, and the gradient

magnitude is used as the mass of the pixels, just like von Gioi *et al.* did. NGC is computed in three steps:

1. Project all the points in the GCA-support region to unit sphere.
2. Compute the initial NGC with plane fitting method.
3. Compensate NGC to satisfy that its associated plane is passing through the center of GCA.

The length and the width are chosen to cover the GCA-support region. Both are angular values since GCA is defined on the unit sphere. Fig. 4 shows an example of the result.

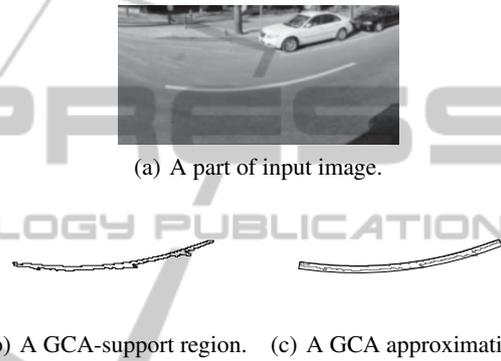


Figure 4: Example of a GCA approximation of a GCA-support region.

3.4 Validation of GCAs

Like von Gioi *et al.*'s, the validation step in our algorithm is also based on the *a contrario* approach and the Helmholtz principle proposed by Desolneux *et al.* (Desolneux *et al.*, 2000; Desolneux *et al.*, 2008). The detection can be considered as a hypothesis testing problem.

There are many tests T_{gca} as there are potential GCAs in the equirectangular image. Similar to rectangle cases, there are $(NM)^2$ potential GCAs, starting and ending on a point of the grid Γ . If we accept \sqrt{NM} possible width values for the GCAs, this gives $(NM)^{\frac{5}{2}}$ tests.

Following the LSD (Gioi *et al.*, 2010), we define the Number of False Alarms of $gca \in T_{gca}$ and an image i , as

$$NFA(gca, i) = \#T_{gca} \cdot P_{H_0}[k(gca, I) \geq k(gca, i)], \quad (3)$$

where P_{H_0} is the probability on the *a contrario* model H_0 , I is a random image on H_0 , $\#T_{gca}$ is the number of potential GCAs in the image, and the statistics $k(gca, x)$ denotes the number of aligned points in the GCA gca and image x . In other words, NFA



Figure 5: Result of our algorithm on internet images (Flicker, 2011).

value is the expected number of detection on the non-structured model. Therefore, the smaller the $NFA(gca, i)$ value, the more meaningful gca is, *i.e.*, the less likely it is to appear in an image drawn from the H_0 . We reject H_0 if and only if $NFA(gca, i) \geq \epsilon$ gives.

The method is justified in the same way as von Gioi *et al.*, *i.e.*, the expected number of detection on the non-structured model is less than ϵ , which gives back the Helmholtz principle. And the dependence of the method on ϵ is very weak. For more details, see (Desolneux *et al.*, 2000). In our experiment, we fix $\epsilon = 1$ once for all.

4 RESULTS

All the experiments were done without tuning any parameter at all. In our experiments, τ is 22.5° , ϵ is 1, and ρ is given as $\frac{q}{\sin \tau}$ where q is the norm of quantization noise (for more details, see (Gioi *et al.*, 2010; Desolneux *et al.*, 2002)).

Fig. 5 shows a series of experiments on the internet images. The images on the left column are the

input images, and the ones on the right column are the results of the proposed algorithm. As you can see in the results, almost all of the expected GCAs are found, except some small ones. And there are very few false detections. Some images contain non-geometrical contents, such as vegetation or cloud, and they do not correspond to real straight or flat objects as shown in the third image of Fig. 5. However, it is a reasonable interpretation in terms of the structure present on the image at a given resolution. The results illustrated in Fig. 5 are acceptable in the sense that every detection corresponds to a locally straight structure in the equirectangular image.

Fig. 6 shows a comparison of our algorithm with Sacht *et al.*'s method for two images. The first row shows input images with magnified region to see the results in detail. The middle and the bottom rows illustrate the results of Sacht *et al.* method and the proposed algorithm, respectively.

In the result of Sacht *et al.*'s method, there are many undetected line segments. In addition, it has much false detection. On the other hand, the result of the proposed algorithm represents well the overall structure of the scene. In addition, the lines that

were not detected in Sacht *et al.*'s were detected well. However, our algorithm also has several false detections. This is because a given image is similar to the tunnel and it contains many curves.

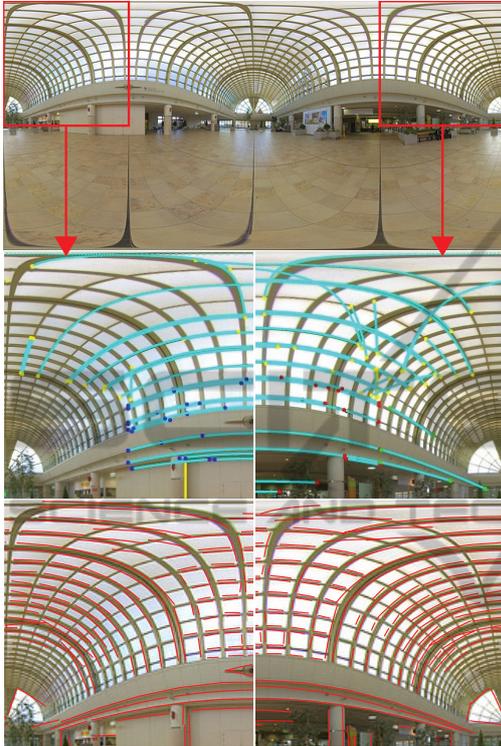


Figure 6: Comparison of two methods: Sacht *et al.* (middle) and the proposed algorithm (bottom).

5 CONCLUSIONS

In this work, we presented a GCA detector in the equirectangular images. The key idea of our algorithm is to represent the line segments in the equirectangular image as GCAs. With this representation, we can define the GCA-support region and approximate geometrical object, which associated with a line segment. Finally, we successfully extend the LSD algorithm, while maintaining most of its advantages. The proposed algorithm gives accurate results and a controlled number of false detections, without any parameter tuning for each image.

Although the proposed algorithm inherits a set of good features from the original LSD method, the computation of $k(gca, i)$ and $n(gca)$ are more complicated than the rectangular case. To make things worse, the effect of noise is also critical for the proposed algorithm as von Gioi *et al.* noted.

To address the limitations of our work, we intent

to carefully consider the validation stage. In addition, the computation of NFA still has to be improved. Thus, quantitative and qualitative analysis of false detection would be done. Also, we will analyze the effect of noise in equirectangular images.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0006132).

REFERENCES

- Ballard, D. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.
- Burns, J. B., Hanson, A. R., and Riseman, E. M. (1986). Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:425–455.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698.
- Desolneux, A., Ladjal, S., Moisan, L., and Morel, J.-M. (2002). Dequantizing image orientation. *IEEE Transactions on Image Processing*, 11(10):1129–1140.
- Desolneux, A., Moisan, L., and Morel, J. (2008). *From Gestalt theory to image analysis: a probabilistic approach*. Interdisciplinary applied mathematics. Springer.
- Desolneux, A., Moisan, L., and Morel, J.-M. (2000). Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23.
- Fiala, M. and Basu, A. (2002). Hough transform for feature detection in panoramic images. *Pattern Recognition Letters*, 23:1863–1874.
- Flicker (2011). Flickr: Equirectangular. <http://www.flickr.com/groups/equirectangular/>.
- Gioi, R. G. V., Jakubowicz, J., Morel, J.-M., and Randall, G. (2008). LSD: A line segment detector. Technical report.
- Gioi, R. G. V., Jakubowicz, J., Morel, J.-M., and Randall, G. (2010). LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:722–732.
- Sacht, L. K., Carvalho, P. C., Velho, L., and Gattass, M. (2010). Face and straight line detection in equirectangular images. In *Proceedings of Workshop of Computer Vision (WVC)*.
- Szenberg, F. (2001). *ACOMPANHAMENTO DE CENAS COMCALIBRAO AUTOMTICA DE CMERAS*. PhD thesis, PUC-Rio. PhD thesis, PUC-Rio.
- Weisstein, E. W. (2011). Equirectangular projection. <http://mathworld.wolfram.com/EquirectangularProjection.html>.